



**BHARATI VIDYAPEETH'S
INSTITUTE OF COMPUTER APPLICATIONS & MANAGEMENT**

(Affiliated to Guru Gobind Singh Indraprastha
University, Approved by AICTE, New Delhi)

Operating Systems with Linux Lab (MCA-163) Practical File

Submitted To:

Dr. Sunil Pratap Singh
(Associate Professor)

Submitted By:

Manpreet Singh (030)
MCA 1st Sem, Sec 1

INDEX

S. No.	Problem Description	Date of Execution	Sign.
P1	Install VirtualBox and then configure Linux (Ubuntu) in VirtualBox.	15/11/2023	
P2	Run ps and note the PID of your shell. Log out and log in again and run ps again. What do you observe?	15/11/2023	
P3	Enter the following commands, and note your observations: (i) who and tty, (ii) tput clear, (iii) id, (iv) ps and echo.	15/11/2023	
P4	Run the following commands, and then invoke ls. What do you conclude? echo > README [Enter] echo > readme [Enter]	15/11/2023	
P5	Create a directory, and change to that directory. Next, create another directory in the new directory, and then change to that directory too. Now, run \$ cd without any arguments followed by pwd. What do you conclude?	15/11/2023	
P6	Create a file mca containing the words "Hello MCA Class!". Now create a directory bvica, and then run mv mca bvica. What do you observe when you run both ls and ls	15/11/2023	

	bar?		
P7	Run \$ who am i and then interpret the output.	15/11/2023	
P8	Find out whether the following commands are internal or external: echo, date, pwd, and ls.	15/11/2023	
P9	Display the current date in the form dd/mm/yyyy.	15/11/2023	
P10	<p>Both of the following commands try to open the file mca, but the error messages are a little different. What could be the reason? (CO2)</p> <p>\$ cat mca</p> <p>cat: mca: No such file or directory</p> <p>\$ cat < mca</p> <p>bash: mca: No such file or directory</p>	15/11/2023	

P1 1	<p>Run the following commands, and discuss their output? (CO2)</p> <ol style="list-style-type: none">1. \$ uname2. \$ passwd3. \$ echo \$SHELL	15/11/2023	
---------	--	------------	--

	<ol style="list-style-type: none">1. \$ man man2. \$ which echo3. \$ type echo4. \$ whereis ls5. \$ cd6. \$ cd \$HOME7. \$ cd ~		
P1 2	<p>Frame ls command to (i) mark directories and executables separately, and (ii) also display hidden files.</p>	15/11/2023	

P1 3	Find out the result of following: \$ cat mca mca mca	15/11/2023	
P1 4	Run the following and determine which commands will work? Explain with reasons. (CO2) 1. \$ mkdir a/b/ 2. \$ mkdir a a/b 3. \$ rmdir a/b/c	15/11/2023	

	1. \$ rmdir a a/b 2. \$ mkdir /bin/mca		
P1 5	How does the command mv mca1 mca2 behave, where both mca1 and mca2 are directories, when (i) mca2 exists, (ii) mca2 doesn't exist?	15/11/2023	

P1 6	<p>Assuming that you are repositioned in the directory /home/bvicam, what are these commands presumed to do, and explain whether they will work at all:</p> <p>(a) \$ cd ../../</p> <ol style="list-style-type: none"> 1. \$ mkdir ../bin 2. \$ rmdir .. 3. \$ ls .. 	17/11/2023	
P1 7	<p>Apply Peterson algorithm for solving the critical section problem with C/Java multi-threaded programming. Assume appropriate code snippet for critical section.</p>	17/11/2023	
P1 8	<p>Apply Bakery algorithm for synchronization of processes/threads in a C/Java program. Assume appropriate code snippet for critical section.</p>	17/11/2023	
P1 9	<p>Write C/Java program to simulate and solve the Producer-</p>	17/11/2023	
	<p>Consumer problem.</p>		

P2 0	Implement Semaphore(s) in aC/Java-multithreaded program to simulate the working and solution of Reader-Writer problem. Assume multiple readers and writers.	17/11/2023	
P2 1	Create a zombie process and an orphan process in a „C“ program with appropriate system calls.	25/11/2023	
P2 2	Write a „C“ program which creates a new process and allows both, child and parent, to report their identification numbers (ids). The parent process should wait for the termination of the child process.	25/11/2023	
P2 3	Write two „C“ programs (A.c and B.c) where one program (A.c) creates a child process and then that child process executes the code of other program (B.c). The logic of program „B.c“ is to generate all the prim number within the specified limit.	25/11/2023	
P2 4	Write an appropriate „C“ program which implements the concept of dynamic memory allocation (use of malloc(), calloc(), realloc(), and free() system call).	25/11/2023	
P2 5	Create a text file, named as „courses.txt“ that contains the	25/11/2023	

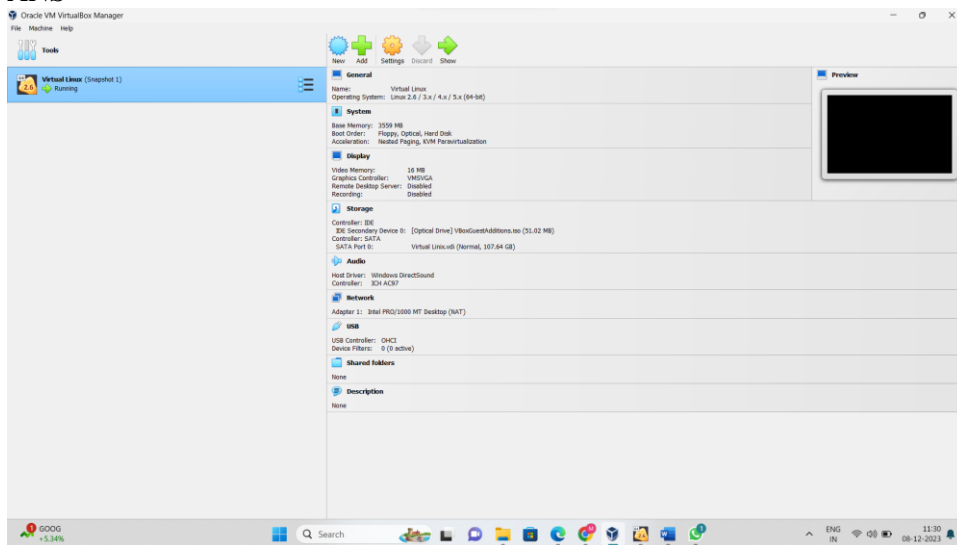
	<p>following four lines:</p> <p>Java Programming Operating System Discrete Structure</p> <p>Write a „C“ program that forks three other processes. After forking, the parent process goes into wait state and waits for the children to finish execution. Each child process reads a line from the „course.txt“ file (Child 1 Reads Line 1, Child 2 Reads Line 2, and Child 3 Reads Line 3) and each prints the respective line. The lines can be printed in any order.</p>		
P2 6	<p>Write a „C“ program (using appropriate system calls of Linux) that generates „n“ integers and stores them in a text file, named as „All.txt“. Then, retrieve the stored integers from this file and copy to “Odd.txt” and „Even.txt“ based upon the type of number, i.e. if the retrieved integer is odd number then store in „Odd.txt“ file or if the retrieved integer is even then store in „Even.txt“ file. Finally, display the contents of all three files on the screen.</p>	01/12/2023	
P2 7	<p>Write a program in „C“ which accepts the file or directory name and permission (access rights) from the user and then changes the access rights accordingly. Use appropriate system call(s) of Linux.</p>	01/12/2023	

P2 8	Write a „C“ program (using appropriate system calls of Linux) which generates and stores the characters from „a“ to „z“. Then, display the stored characters in alternative manner, like: a, c, e, g, ..., etc.	01/12/2023	
---------	---	------------	--

P2 9	Write a „C“ program (using appropriate system calls of Linux) which receives roll number and names of „n“ students, from the user one-by-one and then stores them in a text file, named as „Student.txt“. After inserting all „n“ roll numbers and names, display the contents of file. Also, display the access rights of the file „Student.txt“.	01/12/2023	
P3 0	Demonstrate the use of following system calls by writing an appropriate „C“ program. (CO4) 1. lseek() 2. chmod() 3. umask() 4. access() 5. utime()	01/12/2023	

P1 Install VirtualBox and then configure Linux (Ubuntu) in VirtualBox.

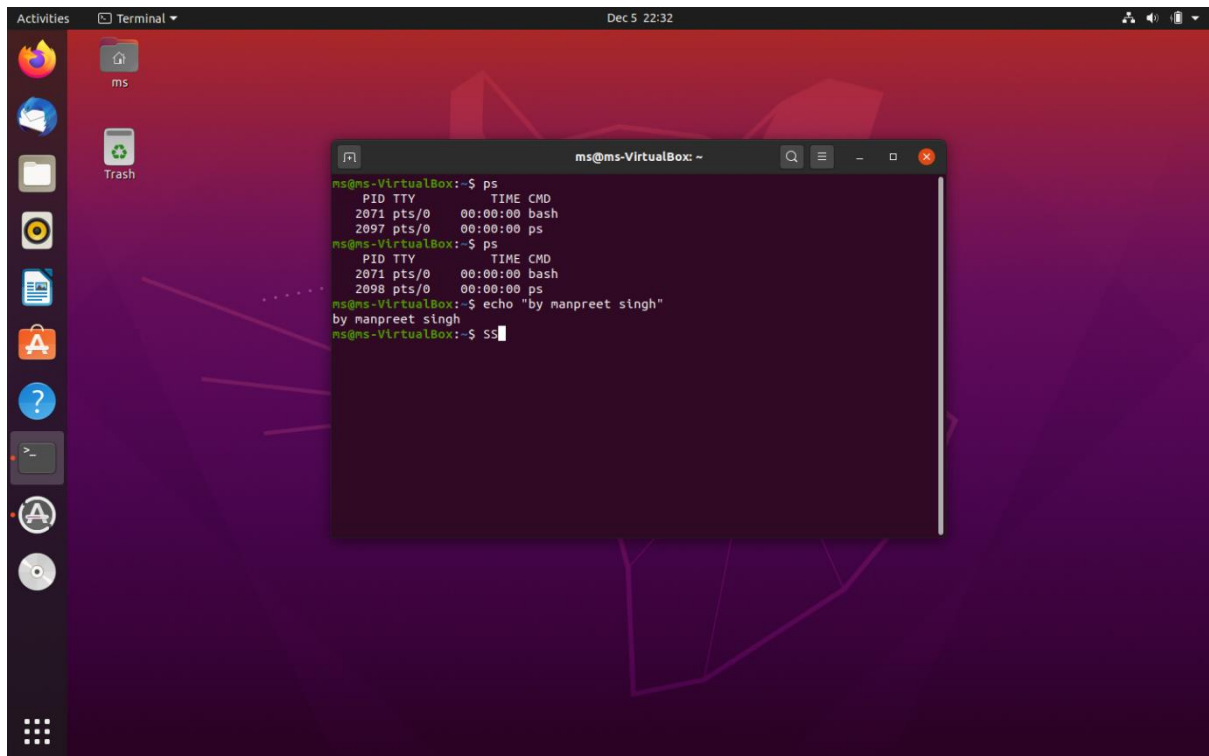
ANS



P2 Run ps and note the PID of your shell. Log out and log in again and run ps again. What do you observe?

ANS

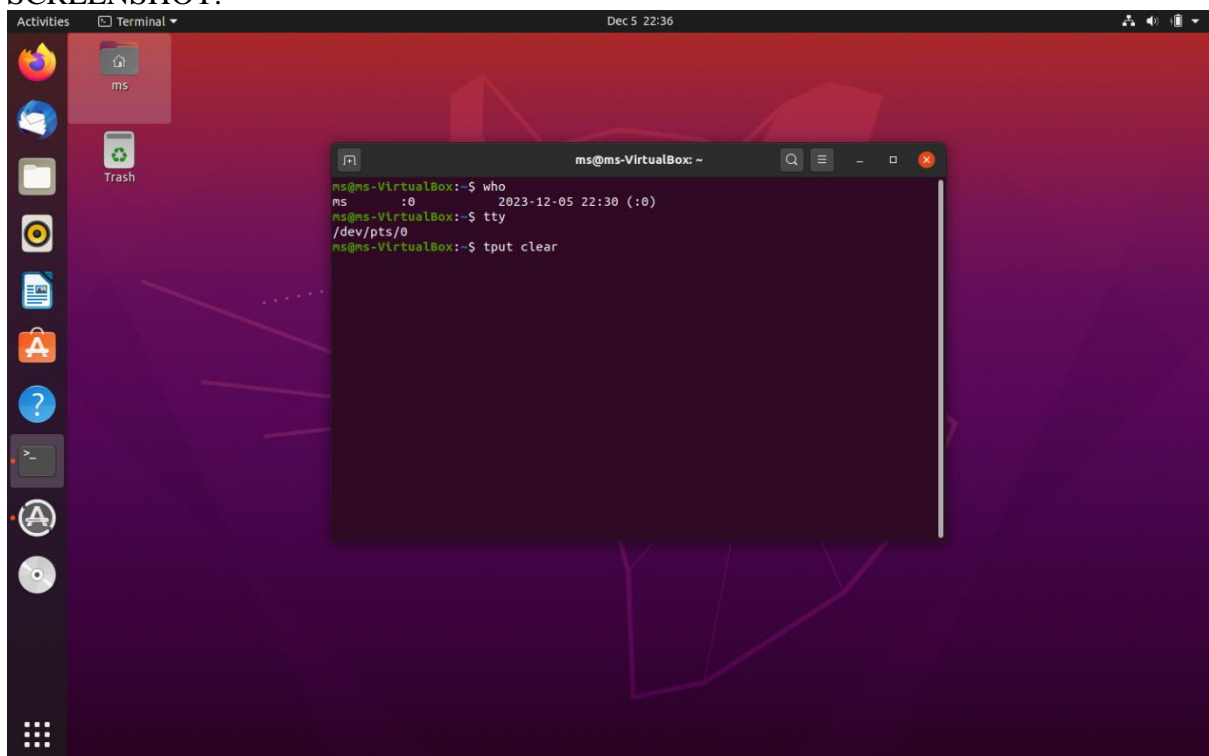
ps command generates important information regarding the tty Such as their pid(s), tty ,time and type of shell Each time we use ps it allocated our bash shell to a new memory location Hence the pids vary when we open and close our bash shell and reopen it.



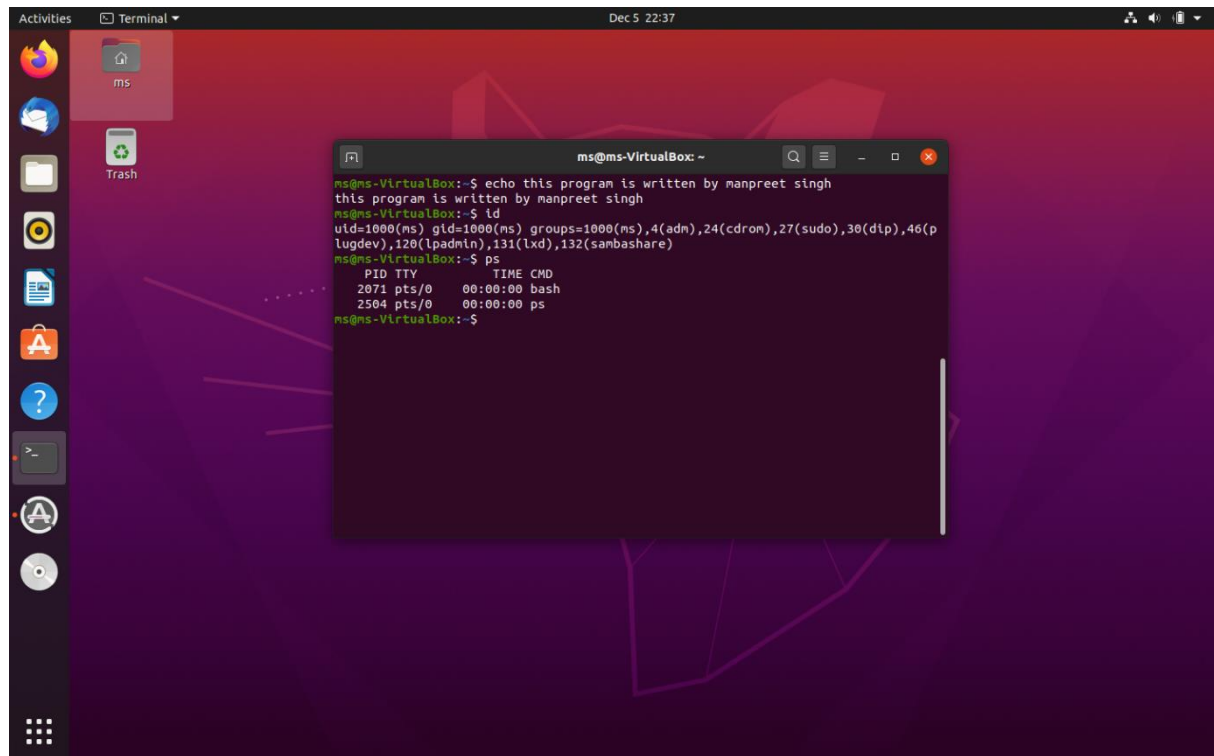
```
ms@ms-VirtualBox:~$ ps
  PID TTY          TIME CMD
 2071 pts/0    00:00:00 bash
 2097 pts/0    00:00:00 ps
ms@ms-VirtualBox:~$ ps
  PID TTY          TIME CMD
 2071 pts/0    00:00:00 bash
 2098 pts/0    00:00:00 ps
ms@ms-VirtualBox:~$ echo "by manpreet singh"
by manpreet singh
ms@ms-VirtualBox:~$ ss
```

P3. Enter the following commands, and note your observations: (i) who and tty, (ii) tput clear, (iii) id, (iv) ps and echo.

SCREENSHOT:



```
ms@ms-VirtualBox:~$ who
ms      :0        2023-12-05 22:30 (:0)
ms@ms-VirtualBox:~$ tty
/dev/pts/0
ms@ms-VirtualBox:~$ tput clear
```

A screenshot of a Linux desktop environment. The desktop has a red and purple gradient background. On the left, there is a vertical dock with icons for Firefox, a file manager, a terminal, and other applications. A terminal window is open in the center, displaying the following commands and their outputs:

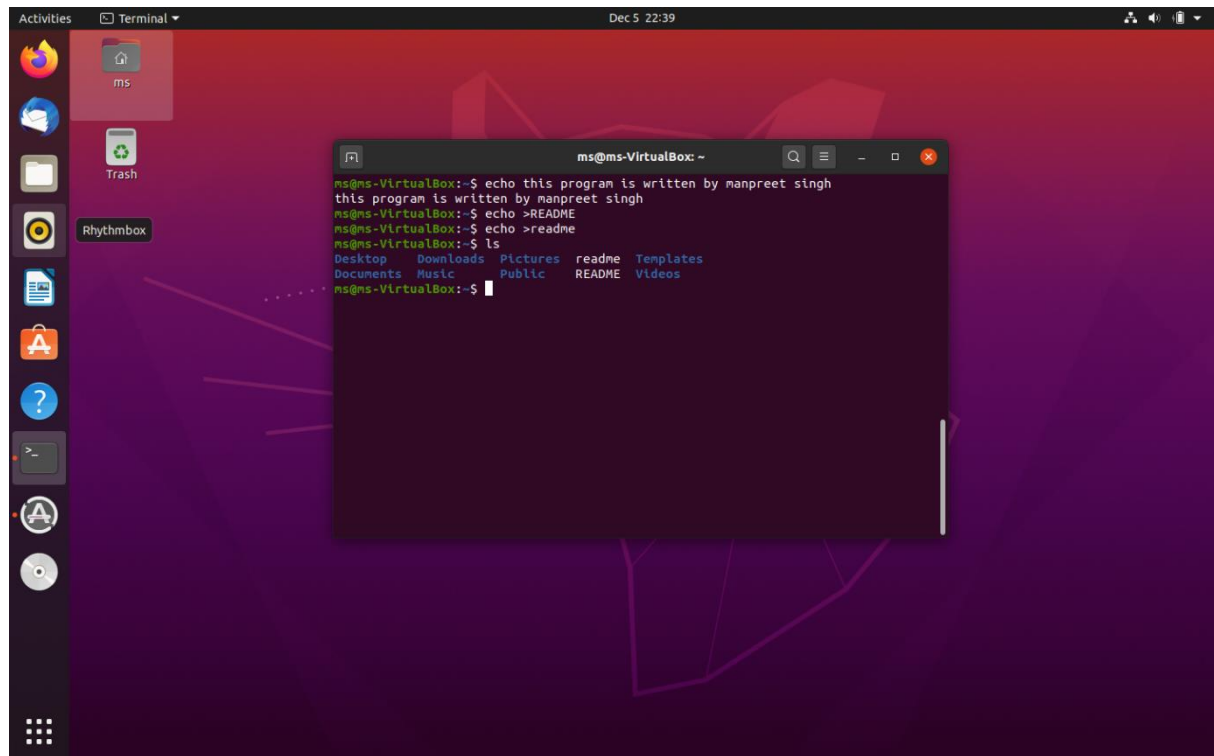
```
ms@ms-VirtualBox:~$ echo this program is written by manpreet singh
this program is written by manpreet singh
ms@ms-VirtualBox:~$ id
uid=1000(ms) gid=1000(ms) groups=1000(ms),4(adm),24(cdrom),27(sudo),30(dip),46(p
lugdev),120(lpadmin),131(lxd),132(sambashare)
ms@ms-VirtualBox:~$ ps
  PID TTY          TIME CMD
 2071 pts/0    00:00:00 bash
 2504 pts/0    00:00:00 ps
ms@ms-VirtualBox:~$
```

P4. Run the following commands, and then invoke ls. What do you conclude?

echo > README [Enter]

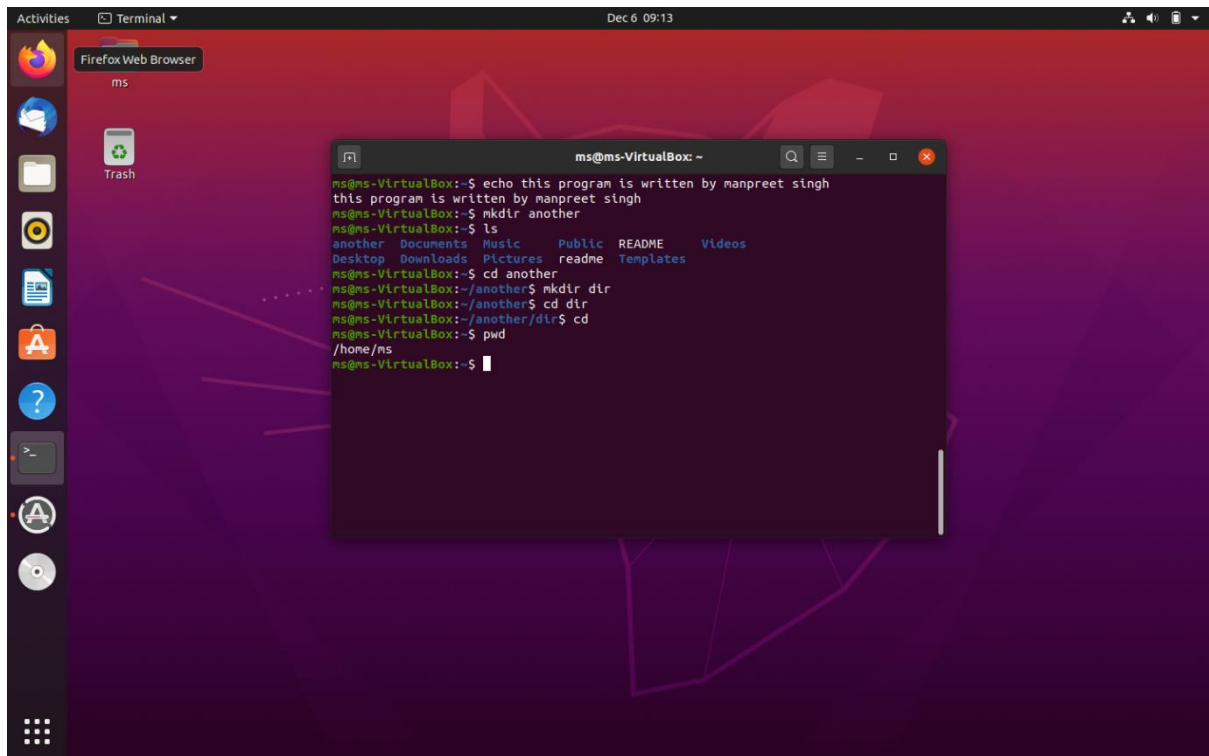
echo > readme [Enter]

SCREENSHOT:



P5. Create a directory, and change to that directory. Next, create another directory in the new directory, and then change to that directory too. Now, run `$ cd` without any arguments followed by `pwd`. What do you conclude?

SCREENSHOT

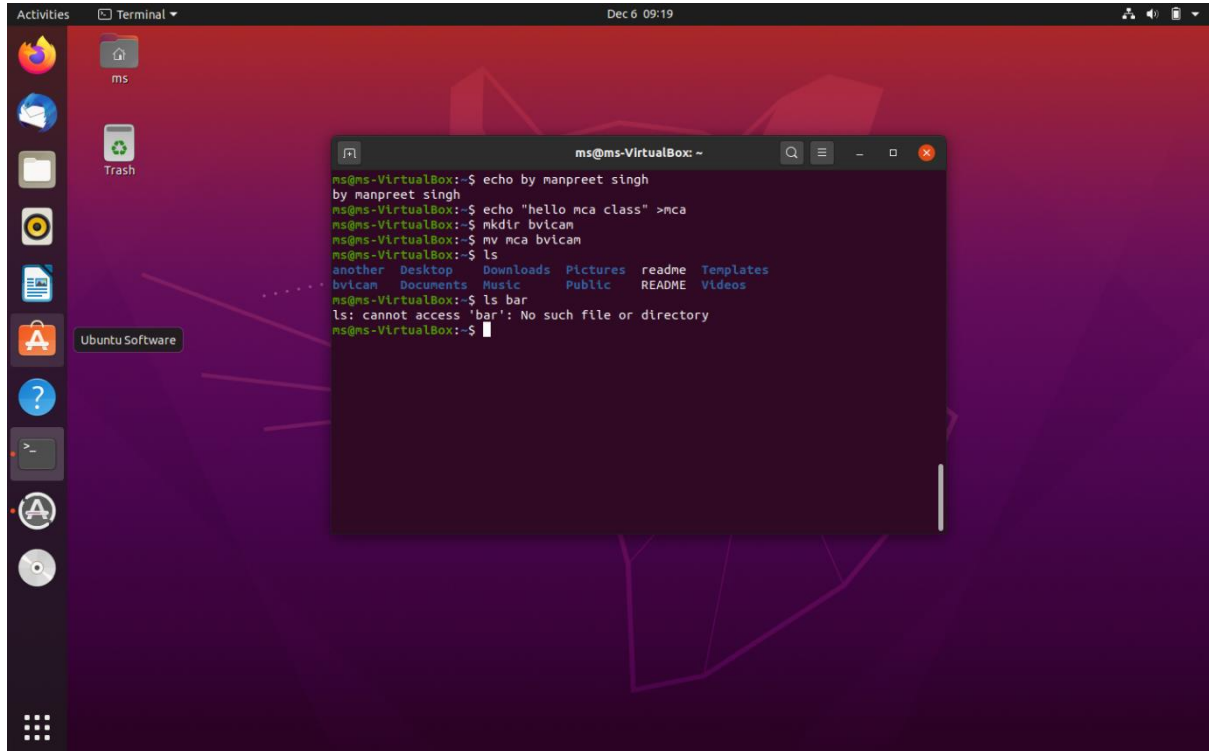


The screenshot shows a Linux desktop with a red and purple background. A terminal window titled 'ms@ms-VirtualBox: ~' is open, displaying the following commands and outputs:

```
ms@ms-VirtualBox:~$ echo this program is written by manpreet singh
this program is written by manpreet singh
ms@ms-VirtualBox:~$ mkdir another
ms@ms-VirtualBox:~$ ls
another Documents Music Public README Videos
Desktop Downloads Pictures readme Templates
ms@ms-VirtualBox:~$ cd another
ms@ms-VirtualBox:~/another$ mkdir dir
ms@ms-VirtualBox:~/another$ cd dir
ms@ms-VirtualBox:~/another/dir$ cd
ms@ms-VirtualBox:~$ pwd
/home/ms
ms@ms-VirtualBox:~$
```

P6 Create a file mca containing the words “Hello MCA Class!”. Now create a directory bvicam, and then run `mv mca bvicam`. What do you observe when you run both `ls` and `ls bar`?

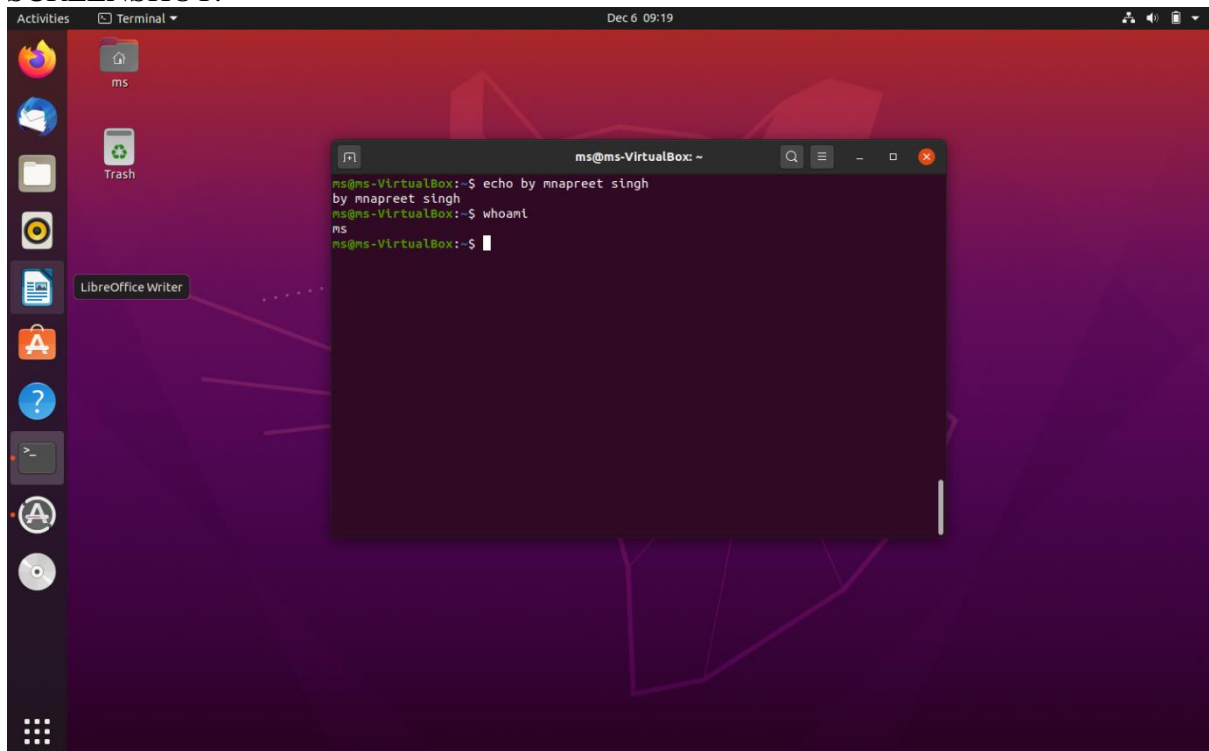
SCREENSHOT:

A screenshot of a Linux desktop environment. The desktop background is a dark purple/red gradient with a faint geometric pattern. On the left side, there is a vertical dock with several application icons: a terminal, a file manager, a web browser, and others. The top of the screen shows a panel with 'Activities', 'Terminal', and the date 'Dec 6 09:19'. A terminal window is open in the center, displaying the following commands and output:

```
ms@ms-VirtualBox:~$ echo by manpreet singh
by manpreet singh
ms@ms-VirtualBox:~$ echo "hello mca class" >mca
ms@ms-VirtualBox:~$ mkdir bvlicam
ms@ms-VirtualBox:~$ mv mca bvlicam
ms@ms-VirtualBox:~$ ls
another Desktop Downloads Pictures readme Templates
bvlicam Documents Music Public README Videos
ms@ms-VirtualBox:~$ ls bar
ls: cannot access 'bar': No such file or directory
ms@ms-VirtualBox:~$
```

P7. Run \$ who am i and then interpret the output.

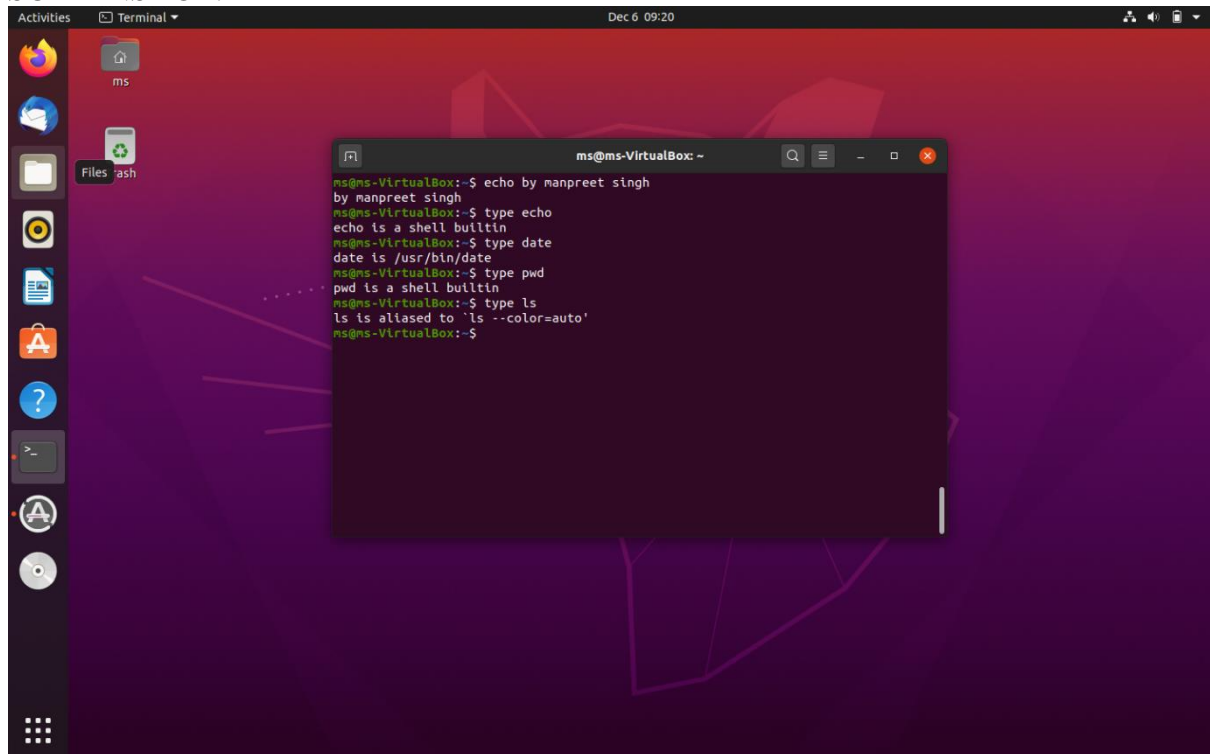
SCREENSHOT:

A screenshot of a Linux desktop environment, similar to the first one. The desktop background is the same dark purple/red gradient. The top panel shows 'Activities', 'Terminal', and the date 'Dec 6 09:19'. A terminal window is open, displaying the following commands and output:

```
ms@ms-VirtualBox:~$ echo by mnapreet singh
by mnapreet singh
ms@ms-VirtualBox:~$ whoami
ms
ms@ms-VirtualBox:~$
```

P8. Find out whether the following commands are internal or external: echo, date, pwd, and ls.

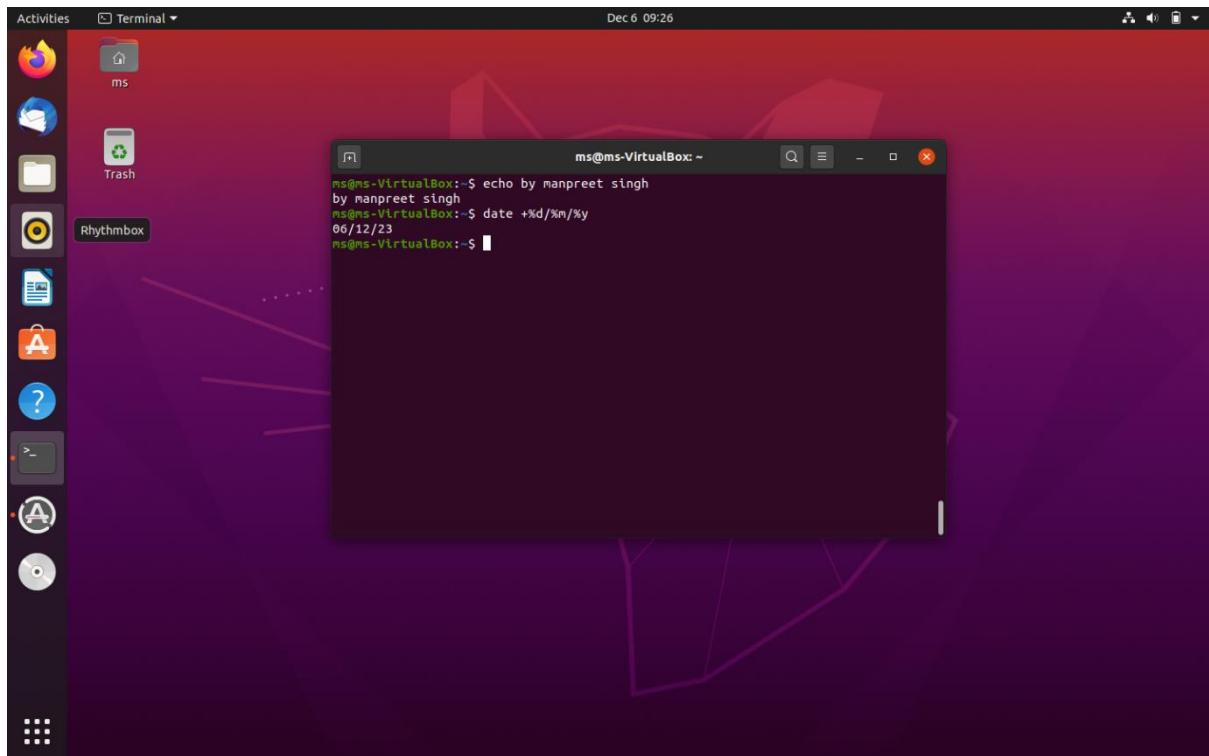
SCREENSHOT:



```
ms@ms-VirtualBox:~$ echo by manpreet singh
by manpreet singh
ms@ms-VirtualBox:~$ type echo
echo is a shell builtin
ms@ms-VirtualBox:~$ type date
date is /usr/bin/date
ms@ms-VirtualBox:~$ type pwd
pwd is a shell builtin
ms@ms-VirtualBox:~$ type ls
ls is aliased to 'ls --color=auto'
ms@ms-VirtualBox:~$
```

P9. Display the current date in the form dd/mm/yyyy.

SCREENSHOT:



P10. Both of the following commands try to open the file mca, but the error messages are a little different. What could be the reason? (CO2)

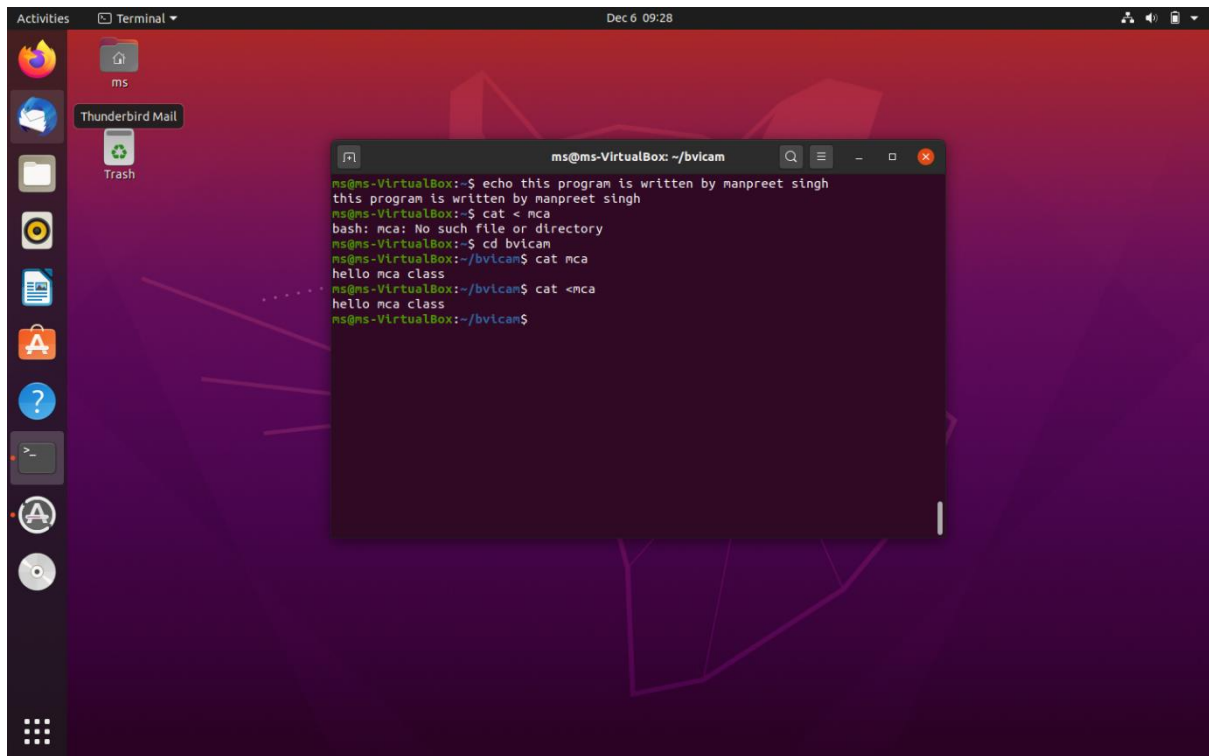
```
$ cat mca
```

```
cat: mca: No such file or directory
```

```
$ cat < mca
```

```
bash: mca: No such file or directory
```

SCREENSHOT:



The screenshot shows a Linux desktop with a purple and red background. A terminal window titled 'ms@ms-VirtualBox: ~/bvican' is open, displaying the following commands and their outputs:

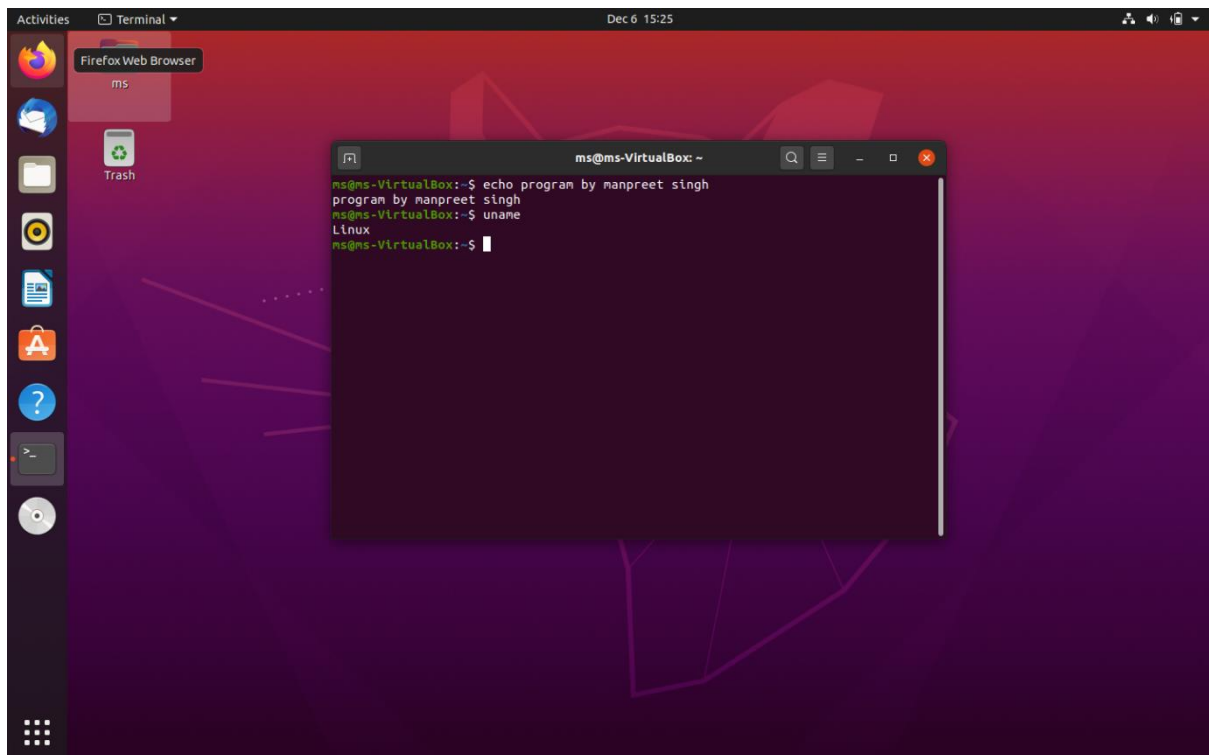
```
ms@ms-VirtualBox:~$ echo this program is written by manpreet singh
this program is written by manpreet singh
ms@ms-VirtualBox:~$ cat < mca
bash: mca: No such file or directory
ms@ms-VirtualBox:~$ cd bvican
ms@ms-VirtualBox:~/bvican$ cat mca
hello mca class
ms@ms-VirtualBox:~/bvican$ cat <mca
hello mca class
ms@ms-VirtualBox:~/bvican$
```

P11. Run the following commands, and discuss their output?

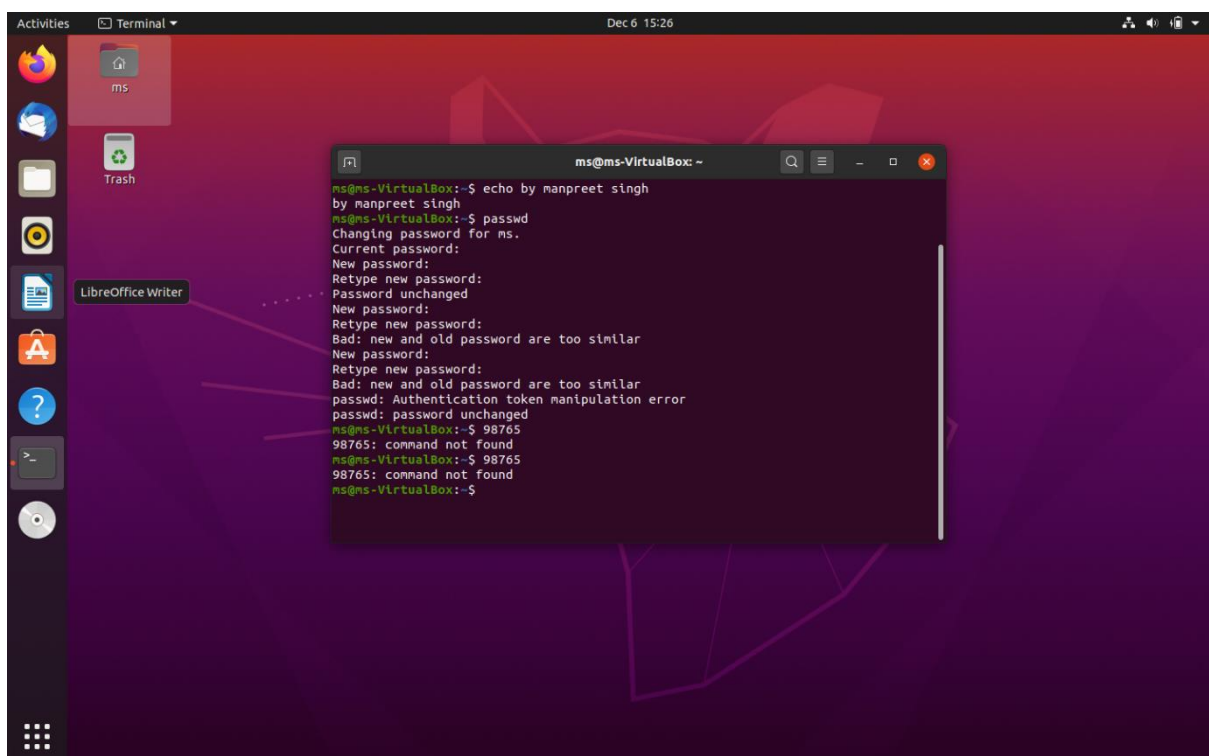
- (a) `$ uname`
- (b) `$ passwd`
- (c) `$ echo $SHELL`
- (d) `$ man man`
- (e) `$ which echo`
- (f) `$ type echo`
- (g) `$ whereis ls`
- (h) `$ cd`
- (i) `$ cd $HOME`
- (j) `$ cd ~-SOLUTION:`

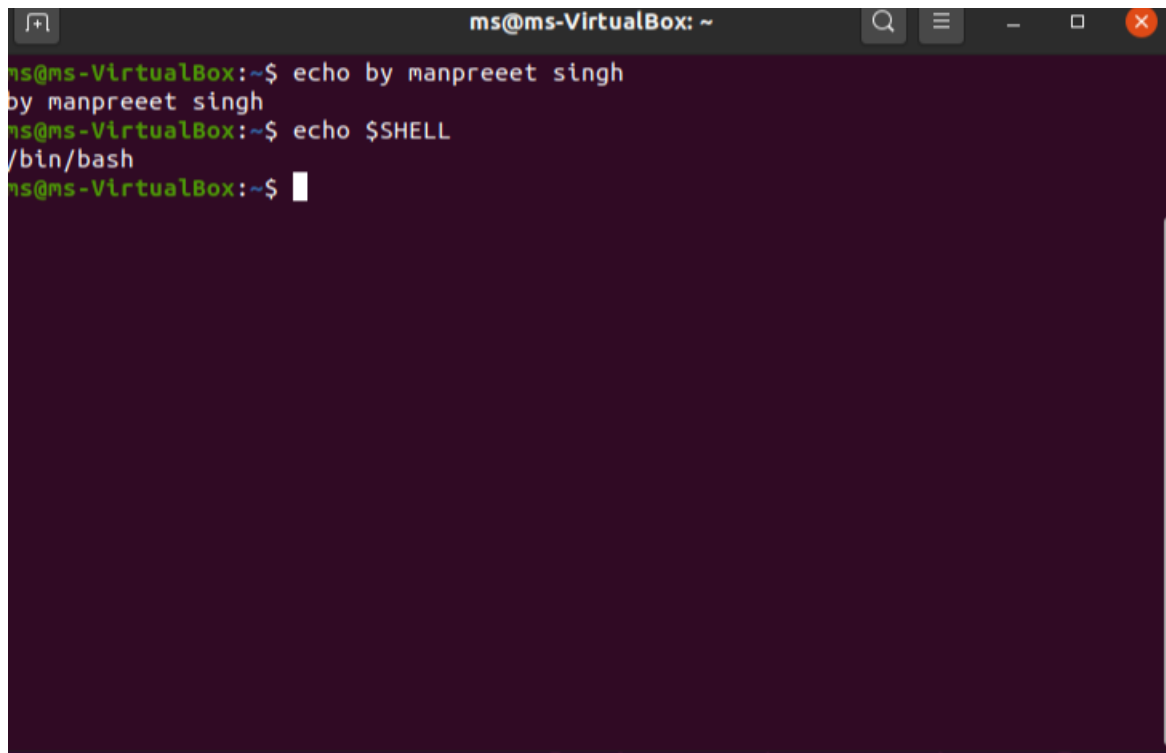
SCREENSHOT:

A

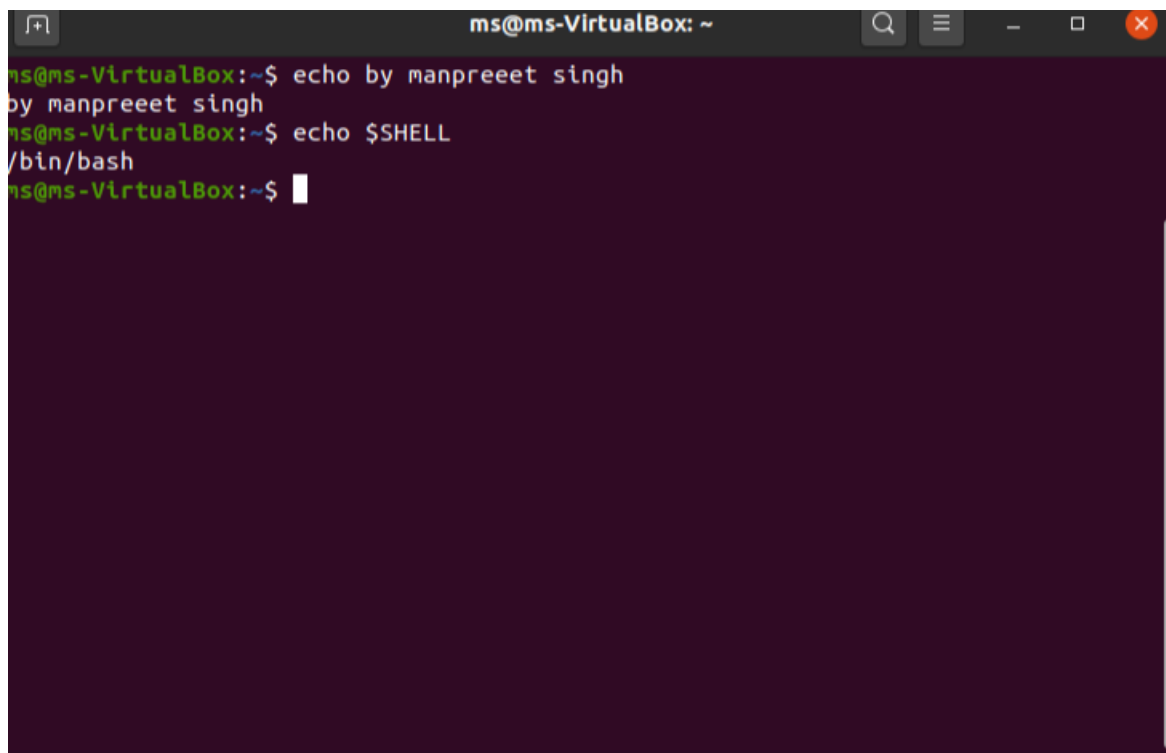


B



CA terminal window titled 'ms@ms-VirtualBox: ~' with standard window controls. The terminal shows the following commands and output:

```
ms@ms-VirtualBox:~$ echo by manpreet singh
by manpreet singh
ms@ms-VirtualBox:~$ echo $SHELL
/bin/bash
ms@ms-VirtualBox:~$
```

DA terminal window titled 'ms@ms-VirtualBox: ~' with standard window controls. The terminal shows the following commands and output:

```
ms@ms-VirtualBox:~$ echo by manpreet singh
by manpreet singh
ms@ms-VirtualBox:~$ echo $SHELL
/bin/bash
ms@ms-VirtualBox:~$
```

E

```
ms@ms-VirtualBox: ~  
ms@ms-VirtualBox:~$ echo by manpreet singh  
by manpreet singh  
ms@ms-VirtualBox:~$ which eco  
ms@ms-VirtualBox:~$ which echo  
/usr/bin/echo  
ms@ms-VirtualBox:~$
```

F

```
ms@ms-VirtualBox: ~  
ms@ms-VirtualBox:~$ echo by manpreet singh  
by manpreet singh  
ms@ms-VirtualBox:~$ type echo  
echo is a shell builtin  
ms@ms-VirtualBox:~$ echo by manpreet singh  
by manpreet singh  
ms@ms-VirtualBox:~$
```

G

```
ms@ms-VirtualBox: ~  
ms@ms-VirtualBox:~$ echo by manpreet singh  
by manpreet singh  
ms@ms-VirtualBox:~$ type echo  
echo is a shell builtin  
ms@ms-VirtualBox:~$ echo by manpreet singh  
by manpreet singh  
ms@ms-VirtualBox:~$ where is ls  
where: command not found  
ms@ms-VirtualBox:~$ whereis ls  
ls: /usr/bin/ls /usr/share/man/man1/ls.1.gz  
ms@ms-VirtualBox:~$
```

H

```
ms@ms-Virtua  
ms@ms-VirtualBox:~$ echo by manpreet singh  
by manpreet singh  
ms@ms-VirtualBox:~$ cd bvica  
ms@ms-VirtualBox:~/bvica$ cd  
ms@ms-VirtualBox:~$
```

J

```
ms@ms-VirtualBox: ~  
ms@ms-VirtualBox:~$ echo by manpreet singh  
by manpreet singh  
ms@ms-VirtualBox:~$ cd bvica  
ms@ms-VirtualBox:~/bvica$ cd $HOME  
ms@ms-VirtualBox:~$
```

J

```
ms@ms-VirtualBox:~$ echo by manpreet singh  
by manpreet singh  
ms@ms-VirtualBox:~$ cd bvica  
ms@ms-VirtualBox:~/bvica$ cd  
ms@ms-VirtualBox:~$
```

P12. Frame ls command to (i) mark directories and executables separately, and (ii) also display hidden files.

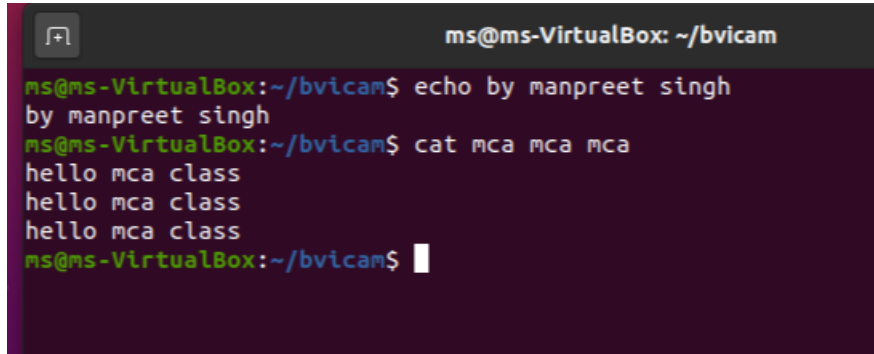
SCREENSHOT:

```
ms@ms-VirtualBox: ~  
ms@ms-VirtualBox:~$ echo by manpreet singh  
by manpreet singh  
ms@ms-VirtualBox:~$ ls -F  
another/ Desktop/ Downloads/ Pictures/ readme Templates/  
bvica/ Documents/ Music/ Public/ README Videos/  
ms@ms-VirtualBox:~$ ls -a  
.  
..  
another  
.bash_history  
.bash_logout  
.bashrc  
bvica  
.cache  
.config  
Desktop  
Documents  
Downloads  
.gnupg  
.local  
.mozilla  
Music  
Pictures  
.profile  
Public  
readme  
README  
.ssh  
.sudo_as_admin_successful  
Templates  
.thunderbird  
.vboxclient-clipboard.pid  
.vboxclient-display-svgx-x11.pid  
.vboxclient-draganddrop.pid  
.vboxclient-seamless.pid  
Videos  
ms@ms-VirtualBox:~$
```

P13. Find out the result of following:

\$ cat mca mca mca

SCREENSHOT:

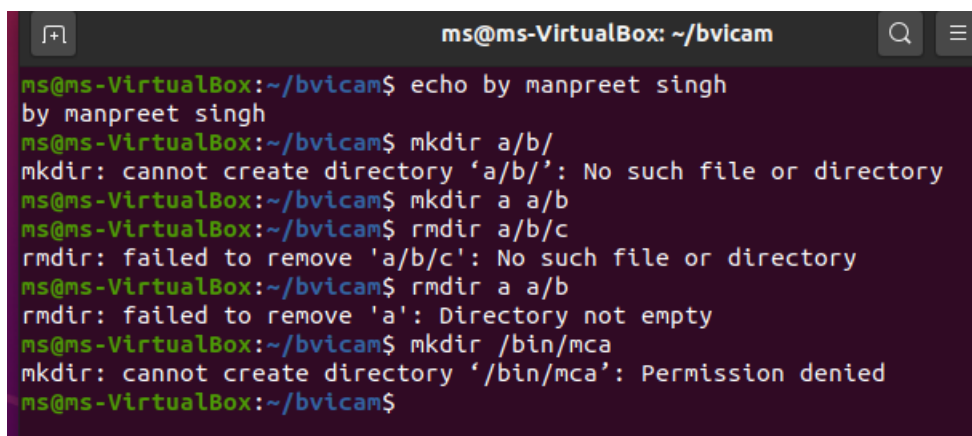
A terminal window titled 'ms@ms-VirtualBox: ~/bvicam' showing the execution of the command 'cat mca mca mca'. The output displays three lines of text: 'hello mca class', 'hello mca class', and 'hello mca class'.

```
ms@ms-VirtualBox: ~/bvicam
ms@ms-VirtualBox:~/bvicam$ echo by manpreet singh
by manpreet singh
ms@ms-VirtualBox:~/bvicam$ cat mca mca mca
hello mca class
hello mca class
hello mca class
ms@ms-VirtualBox:~/bvicam$
```

P14. Run the following and determine which commands will work? Explain with reasons.

- (a) \$ mkdir a/b/
- (b) \$ mkdir a a/b
- (c) \$ rmdir a/b/c
- (d) \$ rmdir a a/b
- (e) \$ mkdir /bin/mca

SCREENSHOT:

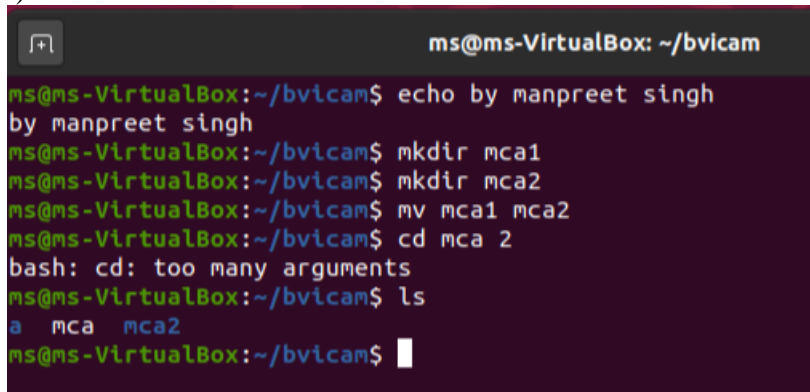
A terminal window titled 'ms@ms-VirtualBox: ~/bvicam' showing the execution of several commands. The commands and their outputs are: 'mkdir a/b/' (successful), 'mkdir a a/b' (error: 'cannot create directory 'a/b/': No such file or directory'), 'rmdir a/b/c' (error: 'failed to remove 'a/b/c': No such file or directory'), 'rmdir a a/b' (error: 'failed to remove 'a': Directory not empty'), and 'mkdir /bin/mca' (error: 'cannot create directory '/bin/mca': Permission denied').

```
ms@ms-VirtualBox:~/bvicam$ echo by manpreet singh
by manpreet singh
ms@ms-VirtualBox:~/bvicam$ mkdir a/b/
ms@ms-VirtualBox:~/bvicam$ mkdir a a/b
mkdir: cannot create directory 'a/b/': No such file or directory
ms@ms-VirtualBox:~/bvicam$ rmdir a/b/c
rmdir: failed to remove 'a/b/c': No such file or directory
ms@ms-VirtualBox:~/bvicam$ rmdir a a/b
rmdir: failed to remove 'a': Directory not empty
ms@ms-VirtualBox:~/bvicam$ mkdir /bin/mca
mkdir: cannot create directory '/bin/mca': Permission denied
ms@ms-VirtualBox:~/bvicam$
```

P15. How does the command `mv mca1 mca2` behave, where both `mca1` and `mca2` are directories, when (i) `mca2` exists, (ii) `mca2` doesn't exist?

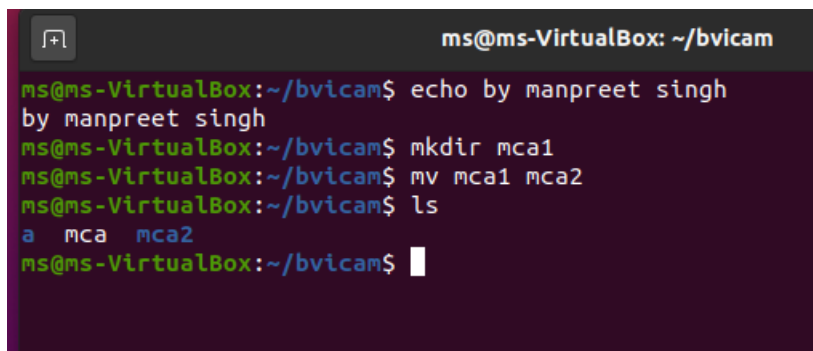
SCREENSHOT:

i)



```
ms@ms-VirtualBox: ~/bvicam
ms@ms-VirtualBox:~/bvicam$ echo by manpreet singh
by manpreet singh
ms@ms-VirtualBox:~/bvicam$ mkdir mca1
ms@ms-VirtualBox:~/bvicam$ mkdir mca2
ms@ms-VirtualBox:~/bvicam$ mv mca1 mca2
ms@ms-VirtualBox:~/bvicam$ cd mca 2
bash: cd: too many arguments
ms@ms-VirtualBox:~/bvicam$ ls
a  mca  mca2
ms@ms-VirtualBox:~/bvicam$
```

2



```
ms@ms-VirtualBox: ~/bvicam
ms@ms-VirtualBox:~/bvicam$ echo by manpreet singh
by manpreet singh
ms@ms-VirtualBox:~/bvicam$ mkdir mca1
ms@ms-VirtualBox:~/bvicam$ mv mca1 mca2
ms@ms-VirtualBox:~/bvicam$ ls
a  mca  mca2
ms@ms-VirtualBox:~/bvicam$
```

P16 Assuming that you are positioned in the directory `/home/bvicam`, what are these commands presumed to do, and explain whether they will work at all:

- (a) `$ cd ../../`
- (b) `$ mkdir ../bin`
- (c) `$ rmdir ..`
- (d) `$ ls ..`

SCREENSHOT:

a)

```
ms@ms-VirtualBox: /home
ms@ms-VirtualBox:~$ echo by manpreet singh
by manpreet singh
ms@ms-VirtualBox:~$ cd bvica
ms@ms-VirtualBox:~/bvica$ pwd
/home/ms/bvica
ms@ms-VirtualBox:~/bvica$ cd ../../
ms@ms-VirtualBox:/home$
```

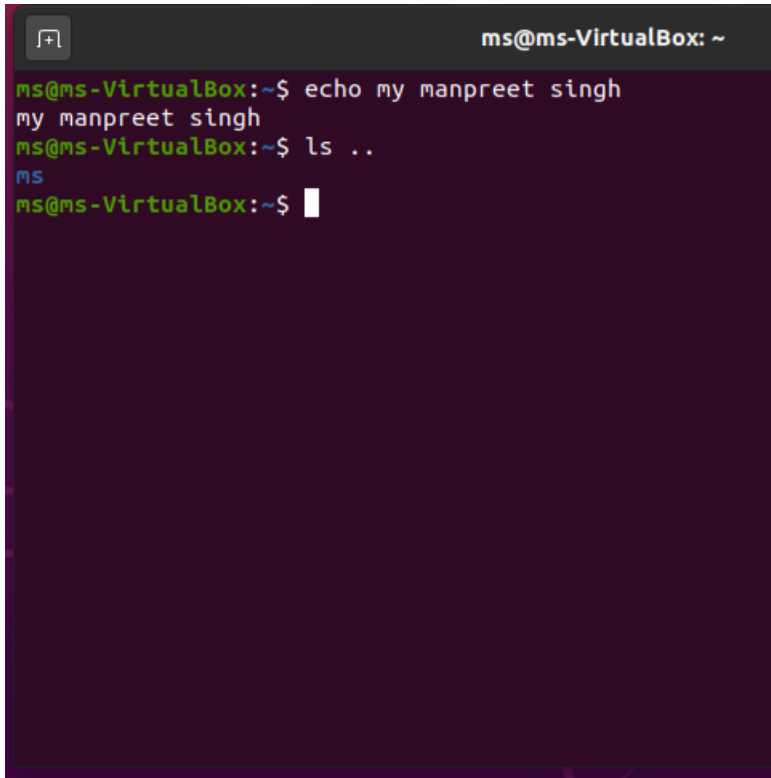
B

```
ms@ms-VirtualBox: ~
ms@ms-VirtualBox:~$ cd bvica
ms@ms-VirtualBox:~/bvica$ ls
a  mca  mca2
ms@ms-VirtualBox:~/bvica$ mkdir ../bin
ms@ms-VirtualBox:~/bvica$ cd
ms@ms-VirtualBox:~$ ls
another  bvica  Documents  java  Pictures  readme  Templates
bin      Desktop  Downloads  Music  Public    README  Videos
ms@ms-VirtualBox:~$ echo my manpreet singh
my manpreet singh
ms@ms-VirtualBox:~$
```

C

```
ms@ms-VirtualBox: ~
ms@ms-VirtualBox:~$ echo by manpreet singh
by manpreet singh
ms@ms-VirtualBox:~$ rmdir ..
rmdir: failed to remove '..': Directory not empty
ms@ms-VirtualBox:~$
```

D



```
ms@ms-VirtualBox: ~  
ms@ms-VirtualBox:~$ echo my manpreet singh  
my manpreet singh  
ms@ms-VirtualBox:~$ ls ..  
ms  
ms@ms-VirtualBox:~$
```

P17. Apply Peterson algorithm for solving the critical section problem with C/Java multi-threaded programming. Assume appropriate code snippet for critical section.

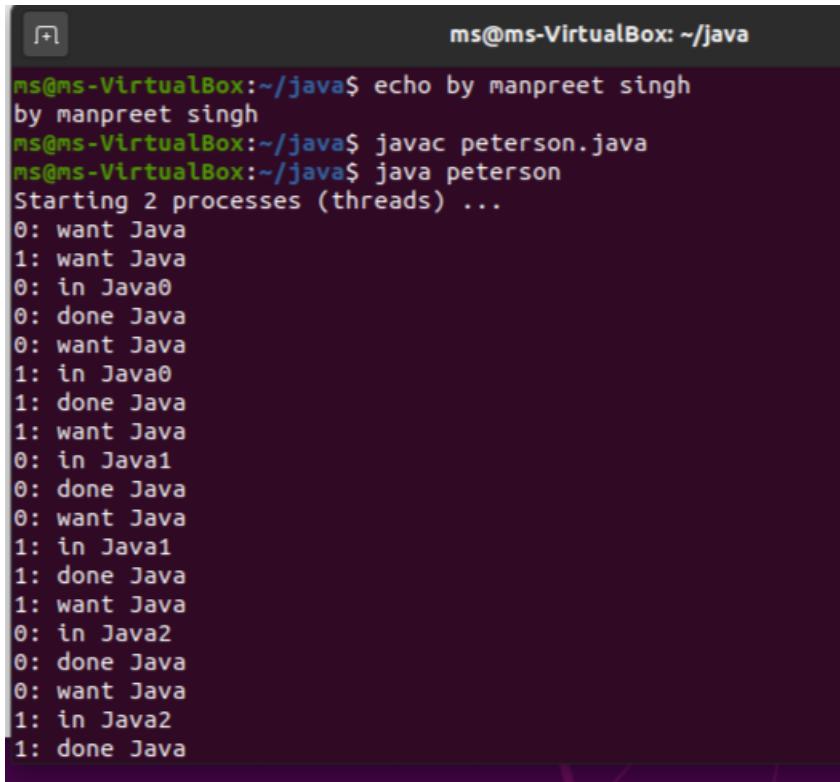
```
class peterson {  
  
    static boolean[] flag = {  
  
        false,  
  
        false  
  
    };  
  
    static int turn = 0;  
  
    static int N = 4;  
  
    static Thread process(int i) {  
  
        return new Thread(() -> {  
  
            int j = 1 - i;  
  

```

```
for (int n = 0; n < N; n++) {  
    log(i + ": want Java"); // LOCK  
    flag[i] = true; // 1  
    turn = j; // 2  
    while (flag[j] && turn == j)  
        Thread.yield(); // 3  
    log(i + ": in Java" + n);  
    sleep(1000 * Math.random()); // 4  
    log(i + ": done Java"); // UNLOCK  
    flag[i] = false; // 5  
}  
});  
}  
  
public static void main(String[] args) {  
    try {  
        log("Starting 2 processes (threads) ...");  
        Thread p0 = process(0);  
        Thread p1 = process(1);  
        p0.start();  
        p1.start();  
        p0.join();  
        p1.join();  
    } catch (InterruptedException e) {}  
}  
  
static void sleep(double t) {  
    try {
```

```
        Thread.sleep((long) t);  
    } catch (InterruptedException e) { }  
}  
  
static void log(String x) {  
    System.out.println(x);  
}  
}
```

SCREENSHOT:



```
ms@ms-VirtualBox: ~/java  
ms@ms-VirtualBox:~/java$ echo by manpreet singh  
by manpreet singh  
ms@ms-VirtualBox:~/java$ javac peterson.java  
ms@ms-VirtualBox:~/java$ java peterson  
Starting 2 processes (threads) ...  
0: want Java  
1: want Java  
0: in Java0  
0: done Java  
0: want Java  
1: in Java0  
1: done Java  
1: want Java  
0: in Java1  
0: done Java  
0: want Java  
1: in Java1  
1: done Java  
1: want Java  
0: in Java2  
0: done Java  
0: want Java  
1: in Java2  
1: done Java
```

P18. Apply Bakery algorithm for synchronization of processes/threads in a C/Java program. Assume appropriate code snippet for critical section.

SOLUTION:

Manpreet singh (03011604423)

```
public class BakeryAlgorithm extends Thread {
    // Variables for the threads.
    public int thread_id; // The id of the current thread.
    public static final int countToThis = 200;
    public static final int numberOfThreads = 5;
    public static volatile int count = 0; // A simple counter for the testing.
    // Global variables for the bakery's algorithm.
    private static volatile boolean[] choosing = new boolean[numberOfThreads];

    private static volatile int[] ticket = new int[numberOfThreads];
    /*
     * Thread constructor.
     */
    public BakeryAlgorithm(int id) {
        thread_id = id;
    }
    // Simple test of a global counter.
    public void run() {
        int scale = 2;
        for (int i = 0; i < countToThis; i++) {
            lock(thread_id);
            // Start of critical section.
            count = count + 1;
            System.out.println("I am " + thread_id + " and count is: " +
                count);

            try {
                sleep((int)(Math.random() * scale));
            } catch (InterruptedException e) {
                /* nothing */
            }
            // End of critical section.
            unlock(thread_id);
        } // for
    } // run method
    /*
     * Method that does the lock of the bakery's algorithm.
     */
    public void lock(int id) {
        choosing[id] = true;
        // Find the max value and add 1 to get the next available ticket.

        ticket[id] = findMax() + 1;
        choosing[id] = false;
        // System.out.println("Thread " + id + " got ticket in Lock");
        for (int j = 0; j < numberOfThreads; j++) {
            // If the thread j is the current thread go the next thread.
            if (j == id)
                continue;
            // Wait if thread j is choosing right now.
            while (choosing[j]) {
                /* nothing */
            }
        }
    }
}
```

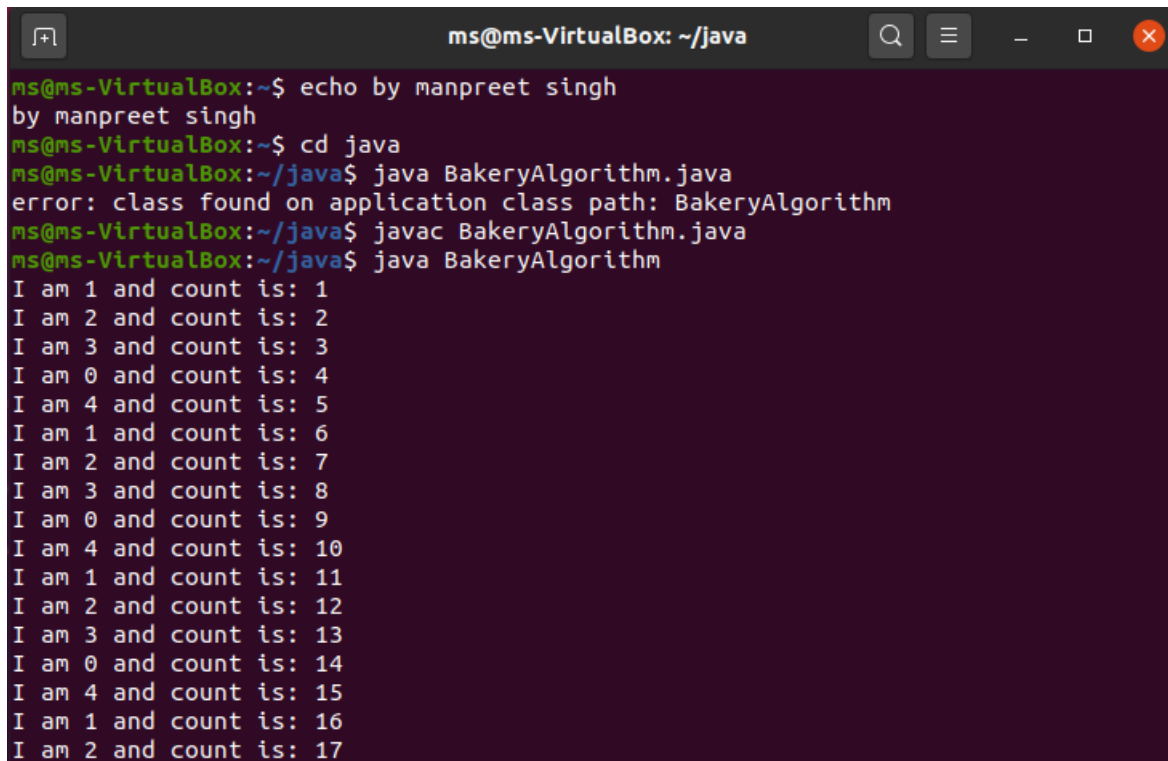
```

        while (ticket[j] != 0 && (ticket[id] > ticket[j] || (ticket[id] ==
            ticket[j] && id > j))) {
            /* nothing */ }
    } // for
}
/*
 * Method that leaves the lock.
 */
private void unlock(int id) {
    ticket[id] = 0;
    // System.out.println("Thread " + id + " unlock");
}
/*
 * Method that finds the max value inside the ticket array.
 */
private int findMax() {
    int m = ticket[0];
    for (int i = 1; i < ticket.length; i++) {
        if (ticket[i] > m)
            m = ticket[i];
    }
    return m;
}
public static void main(String[] args) {
    // Initialization of the global variables (it is not necessary at all).
    for (int i = 0; i < numberOfThreads; i++) {
        choosing[i] = false;
        ticket[i] = 0;
    }
    BakeryAlgorithm[] threads = new BakeryAlgorithm[numberOfThreads]; // Array of
threads.
    // Initialize the threads.
    for (int i = 0; i < threads.length; i++) {
        threads[i] = new BakeryAlgorithm(i);
        threads[i].start();
    }
    // Wait all threads to finish.
    for (int i = 0; i < threads.length; i++) {
        try {

            threads[i].join();
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
    System.out.println("\nCount is: " + count);
    System.out.println("\nExpected was: " + (countToThis *
        numberOfThreads));
}
}

```

Screenshot-

A screenshot of a terminal window titled 'ms@ms-VirtualBox: ~/java'. The terminal shows the following commands and output:

```
ms@ms-VirtualBox:~$ echo by manpreet singh
by manpreet singh
ms@ms-VirtualBox:~$ cd java
ms@ms-VirtualBox:~/java$ java BakeryAlgorithm.java
error: class found on application class path: BakeryAlgorithm
ms@ms-VirtualBox:~/java$ javac BakeryAlgorithm.java
ms@ms-VirtualBox:~/java$ java BakeryAlgorithm
I am 1 and count is: 1
I am 2 and count is: 2
I am 3 and count is: 3
I am 0 and count is: 4
I am 4 and count is: 5
I am 1 and count is: 6
I am 2 and count is: 7
I am 3 and count is: 8
I am 0 and count is: 9
I am 4 and count is: 10
I am 1 and count is: 11
I am 2 and count is: 12
I am 3 and count is: 13
I am 0 and count is: 14
I am 4 and count is: 15
I am 1 and count is: 16
I am 2 and count is: 17
```

P19. Write C/Java program to simulate and solve the Producer-Consumer problem.

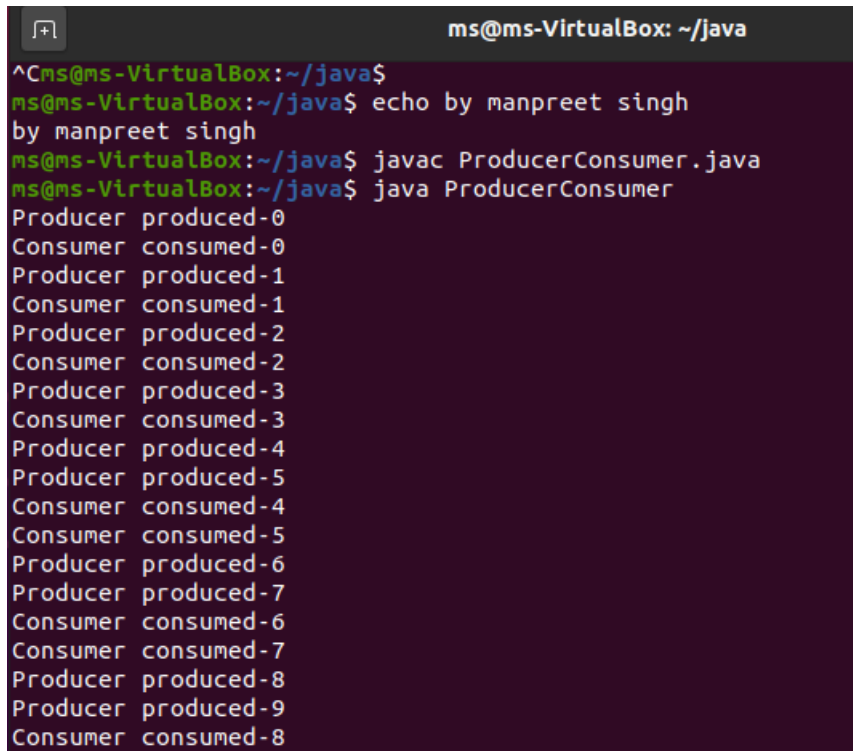
SOLUTION:

```
import java.util.LinkedList;
public class ProducerConsumer {
    public static void main(String[] args)
        throws InterruptedException {
        final PC pc = new PC();
        Thread t1 = new Thread(new Runnable() {
            @Override
            public void run() {
                try {
                    pc.produce();
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
        });
        // Create consumer thread
        Thread t2 = new Thread(new Runnable() {
            @Override
            public void run() {
                try {
                    pc.consume();
                } catch (InterruptedException e) {
```

Manpreet singh (03011604423)

```
        e.printStackTrace();
    }
}
});
// Start both threads
t1.start();
t2.start();
// t1 finishes before t2
t1.join();
```

Screenshot-



The screenshot shows a terminal window titled "ms@ms-VirtualBox: ~/java". The user enters the following commands and receives the following output:

```
ms@ms-VirtualBox:~/java$ echo by manpreet singh
by manpreet singh
ms@ms-VirtualBox:~/java$ javac ProducerConsumer.java
ms@ms-VirtualBox:~/java$ java ProducerConsumer
Producer produced-0
Consumer consumed-0
Producer produced-1
Consumer consumed-1
Producer produced-2
Consumer consumed-2
Producer produced-3
Consumer consumed-3
Producer produced-4
Producer produced-5
Consumer consumed-4
Consumer consumed-5
Producer produced-6
Producer produced-7
Consumer consumed-6
Consumer consumed-7
Producer produced-8
Producer produced-9
Consumer consumed-8
```

P20. Implement Semaphore(s) in a C/Java-multithreaded program to simulate the working and solution of Reader-Writer problem. Assume multiple readers and writers.

SOLUTION:

```
import java.util.concurrent.*;

class Shared {

    static int count = 0;
```



```
}  
  
class MyThread extends Thread {  
  
    Semaphore sem;  
  
    String threadName;  
  
    public MyThread(Semaphore sem, String threadName) {  
  
        super(threadName);  
  
        this.sem = sem;  
  
        this.threadName = threadName;  
  
    }  
  
    @Override  
  
    public void run() {  
  
        // run by thread A  
  
        if (this.getName().equals("A")) {  
  
            System.out.println("Starting " + threadName);  
  
            try {  
  
                // First, get a permit.  
  
                System.out.println(threadName + " is waiting for a permit.");  
  
                // acquiring the lock  
  
                sem.acquire();  
  
                System.out.println(threadName + " gets a permit.");  
  
                for (int i = 0; i < 5; i++) {  
  
                    Shared.count++;  
  
                    System.out.println(threadName + ": " + Shared.count);  
  
                    // Now, allowing a context switch -- if possible.  
  
                    // for thread B to execute  
  
                    Thread.sleep(10);  

```

```
    }

    } catch (InterruptedException exc) {

        System.out.println(exc);

    }

    // Release the permit.

    System.out.println(threadName + " releases the permit.");

    sem.release();

}

// run by thread B

else {

    System.out.println("Starting " + threadName);

    try {

        // First, get a permit.

        System.out.println(threadName + " is waiting for a permit.");

        // acquiring the lock

        sem.acquire();

        System.out.println(threadName + " gets a permit.");

        // Now, accessing the shared resource.

        // other waiting threads will wait, until this

        // thread release the lock

        for (int i = 0; i < 5; i++) {

            Shared.count--;

            System.out.println(threadName + ": " + Shared.count);

            // Now, allowing a context switch -- if possible.

            // for thread A to execute
```

```
        Thread.sleep(10);
    }
} catch (InterruptedException exc) {
    System.out.println(exc);
}

// Release the permit.

System.out.println(threadName + " releases the permit.");
sem.release();
}
}
}

// Driver class

public class SemaphoreDemo {

    public static void main(String args[]) throws InterruptedException {

        // creating a Semaphore object

        // with number of permits 1

        Semaphore sem = new Semaphore(1);

        // creating two threads with name A and B

        // Note that thread A will increment the count

        // and thread B will decrement the count

        MyThread t1 = new MyThread(sem, "A");

        MyThread t2 = new MyThread(sem, "B");

        // stating threads A and B

        t1.start();

        t2.start();

        // waiting for threads A and B
    }
}
```

Manpreet singh (03011604423)

```
t1.join();

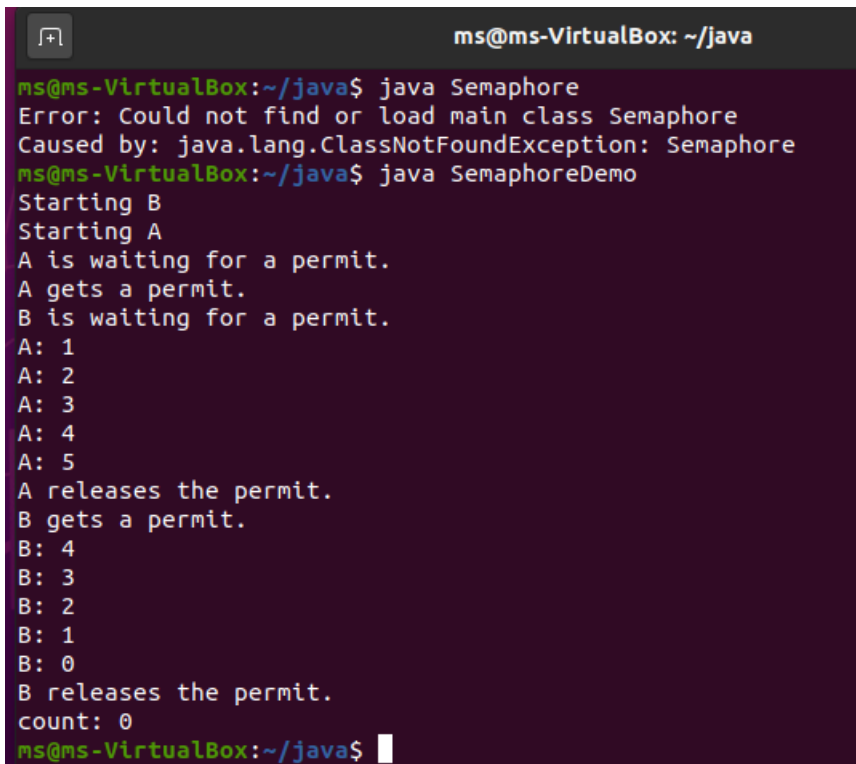
t2.join();

System.out.println("count: " + Shared.count);

}

}
```

Screenshot-



```
ms@ms-VirtualBox: ~/java
ms@ms-VirtualBox:~/java$ java Semaphore
Error: Could not find or load main class Semaphore
Caused by: java.lang.ClassNotFoundException: Semaphore
ms@ms-VirtualBox:~/java$ java SemaphoreDemo
Starting B
Starting A
A is waiting for a permit.
A gets a permit.
B is waiting for a permit.
A: 1
A: 2
A: 3
A: 4
A: 5
A releases the permit.
B gets a permit.
B: 4
B: 3
B: 2
B: 1
B: 0
B releases the permit.
count: 0
ms@ms-VirtualBox:~/java$
```

P21 Create a zombie process and an orphan process in a C program with appropriate system calls.

SOLUTION:

Zombie Process:

```
#include <stdlib.h>
```

```
#include <sys/types.h>
```

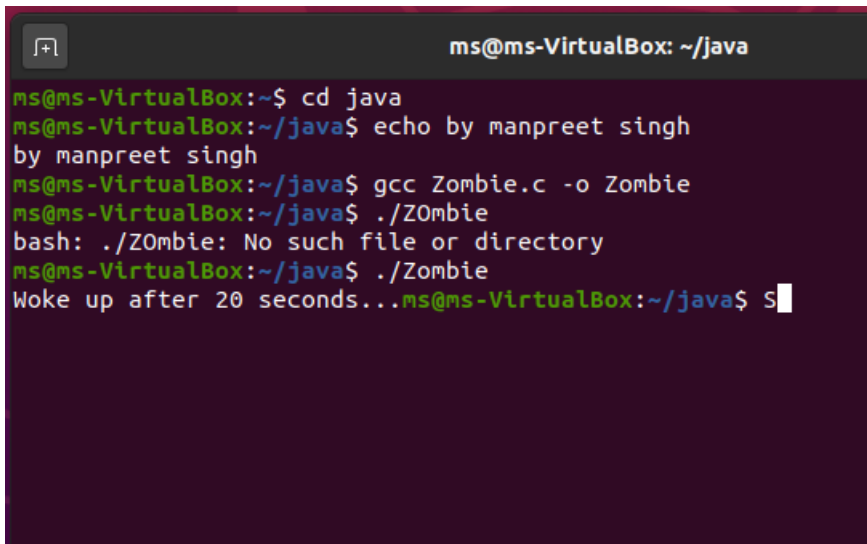
```
#include <unistd.h>
```

Manpreet singh (03011604423)

```
#include<stdio.h>
```

```
int main() {  
    // Fork returns process id  
    // in parent process  
    pid_t child_pid = fork();  
    // Parent process  
    if (child_pid > 0)  
        sleep(20);  
    // Child process  
    else  
        exit(0);  
    printf("Woke up after 20 seconds...");  
    return 0;  
}
```

SCREENSHOT:



```
ms@ms-VirtualBox: ~/java  
ms@ms-VirtualBox:~$ cd java  
ms@ms-VirtualBox:~/java$ echo by manpreet singh  
by manpreet singh  
ms@ms-VirtualBox:~/java$ gcc Zombie.c -o Zombie  
ms@ms-VirtualBox:~/java$ ./Zombie  
bash: ./Zombie: No such file or directory  
ms@ms-VirtualBox:~/java$ ./Zombie  
Woke up after 20 seconds...ms@ms-VirtualBox:~/java$ S
```

Orphan Process

```
#include <stdlib.h>
```

```
#include <sys/types.h>
```

```
#include <unistd.h>
```

```
#include<stdio.h>
```

```
int main()
```

```
{
```

Manpreet singh (03011604423)

```
// Fork returns process id
// in parent process
pid_t child_pid = fork();

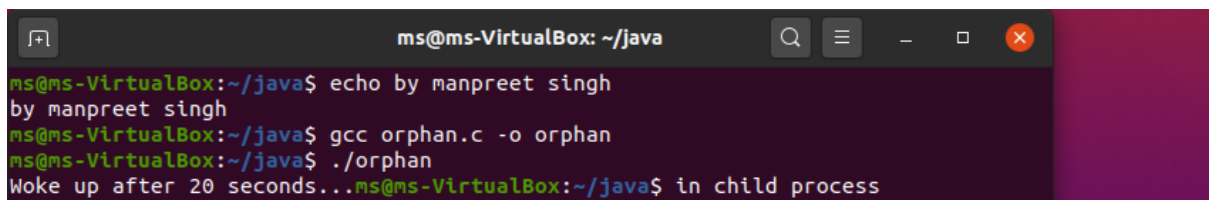
// Parent process
if (child_pid > 0)
    sleep(20);

// Child process
else
    exit(0);

printf("Woke up after 20 seconds...");

return 0;
}
```

SCREENSHOT:



```
ms@ms-VirtualBox: ~/java
ms@ms-VirtualBox:~/java$ echo by manpreet singh
by manpreet singh
ms@ms-VirtualBox:~/java$ gcc orphan.c -o orphan
ms@ms-VirtualBox:~/java$ ./orphan
Woke up after 20 seconds...ms@ms-VirtualBox:~/java$ in child process
```

P22. Write a „C“ program which creates a new process and allows both, child and parent, to report their identification numbers (ids). The parent process should wait for the termination of the child process.

SOLUTION:

```
#include<stdio.h>

#include<fcntl.h>
```

```
#include<stdlib.h>

#include<unistd.h>

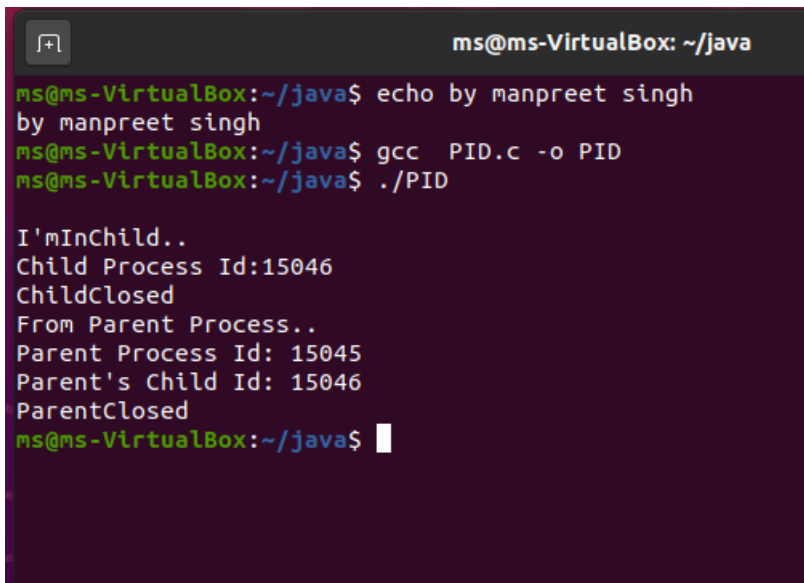
#include<sys/types.h>

#include<sys/stat.h>

#include<sys/wait.h>

int main() {
    pid_t pid;
    pid = fork();
    if (pid == 0) {
        printf("\nI'mInChild..");
        printf("\nChild Process Id:%d", getpid());
        printf("\nChildClosed");
        exit(0);
    } else {
        wait(NULL);
        printf("\nFrom Parent Process..");
        printf("\nParent Process Id: %d", getpid());
        printf("\nParent's Child Id: %d", pid);
        printf("\nParentClosed");
    }
    printf("\n");
    return 0;
}
```

SCREENSHOT:



```
ms@ms-VirtualBox: ~/java
ms@ms-VirtualBox:~/java$ echo by manpreet singh
by manpreet singh
ms@ms-VirtualBox:~/java$ gcc PID.c -o PID
ms@ms-VirtualBox:~/java$ ./PID

I'mInChild..
Child Process Id:15046
ChildClosed
From Parent Process..
Parent Process Id: 15045
Parent's Child Id: 15046
ParentClosed
ms@ms-VirtualBox:~/java$
```

P23. Write two „C“ programs (A.c and B.c) where one program (A.c) creates a child process and then that child process executes the code of

other program (B.c). The logic of program „B.c“ is to generate all the prime numbers within the specified limit.

SOLUTION:

A.c

```
#include<stdio.h>

#include<unistd.h>

int main(){

printf("I am executing A.c \n");

printf("PID of A.c is %d\n",getpid());

char *args[]={ "/B.c",NULL };

execv(args[0],args);

}
```

B.c

```
#include<unistd.h>

#include<sys/types.h>

#include<stdlib.h>

#include<stdio.h>

#include<fcntl.h>

void main()

{

int n;

printf("\nUp to How Many Numbers:");

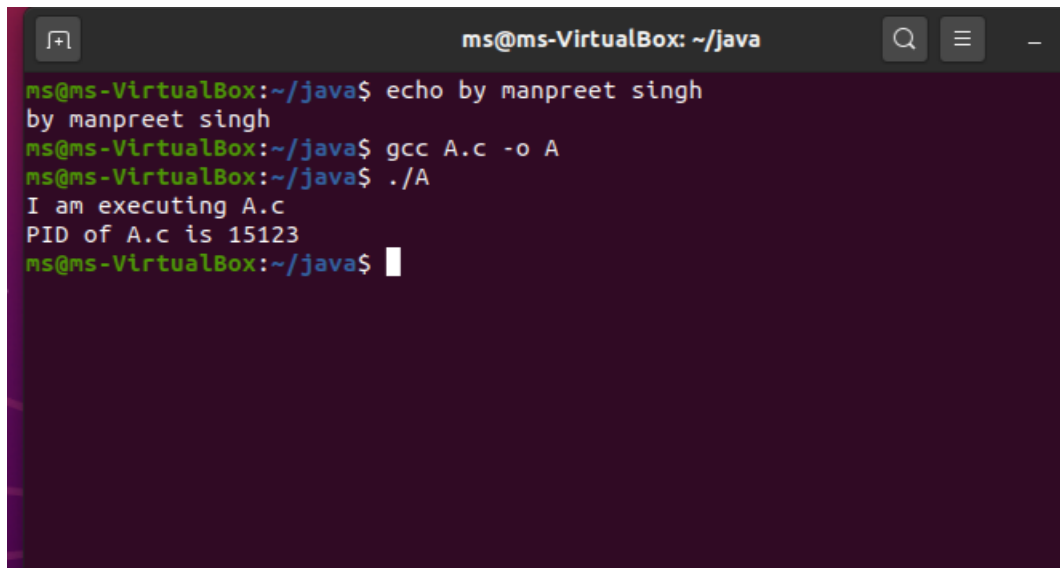
scanf("%d",&n);

for(int i=1; i<n; i++)
```



```
{  
int flag=0;  
for(int j=2; j<=i/2; j++)  
{  
if(i%j==0)  
{  
flag=1;  
break;  
}  
}  
if(i==1)  
{  
printf("\n1 is neither Prime nor Composite..\n");  
}  
else  
{  
if(flag==0)  
{  
printf("%d is a Prime Number..\n",i);  
}  
}  
}  
}
```

SCrenshot-



```
ms@ms-VirtualBox: ~/java
ms@ms-VirtualBox:~/java$ echo by manpreet singh
by manpreet singh
ms@ms-VirtualBox:~/java$ gcc A.c -o A
ms@ms-VirtualBox:~/java$ ./A
I am executing A.c
PID of A.c is 15123
ms@ms-VirtualBox:~/java$
```

P24. Write an appropriate „C“ program which implements the concept of dynamic memory allocation (use of malloc(), calloc(), realloc(), and free() system call).

SOLUTION:

Malloc:

```
#include<stdio.h> #include<malloc.h> #include<stdlib.h> void main()
{
int n, *ptr, i;
printf("Input array size: "); scanf("%d",&n);
ptr = (int *)malloc(n*sizeof(int)); if(ptr==NULL)
{
printf("\nNo Allocation of memory");
}
else
{
printf("\nMemory Allocation Done!"); printf("\nAddress of first byte = %p", ptr); for(i=0;
i<n; i++)
{
Manpreet singh (03011604423)
```

```
ptr[i] = i+10;

}

}

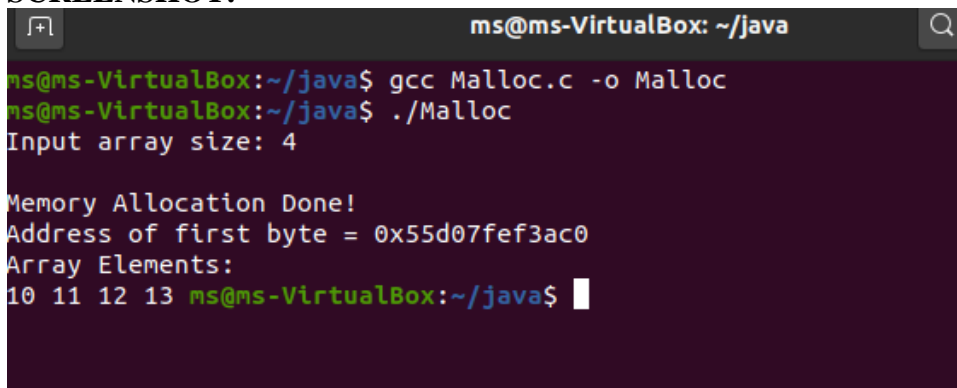
printf("\nArray Elements: \n"); for(i=0; i<n; i++)

{

printf("%d ", ptr[i]);

// printf("%p ", ptr+i);

}
```

SCREENSHOT:

```
ms@ms-VirtualBox: ~/java
ms@ms-VirtualBox:~/java$ gcc Malloc.c -o Malloc
ms@ms-VirtualBox:~/java$ ./Malloc
Input array size: 4

Memory Allocation Done!
Address of first byte = 0x55d07fef3ac0
Array Elements:
10 11 12 13 ms@ms-VirtualBox:~/java$
```

[Calloc:](#)

```
#include <stdio.h> #include <stdlib.h> int main()

{

int* ptr; int n, i; n = 5;

printf("Number of elements: %d\n", n);

// Dynamically allocate memory using calloc() ptr = (int*)calloc(n, sizeof(int));

// Check if the memory has been successfully

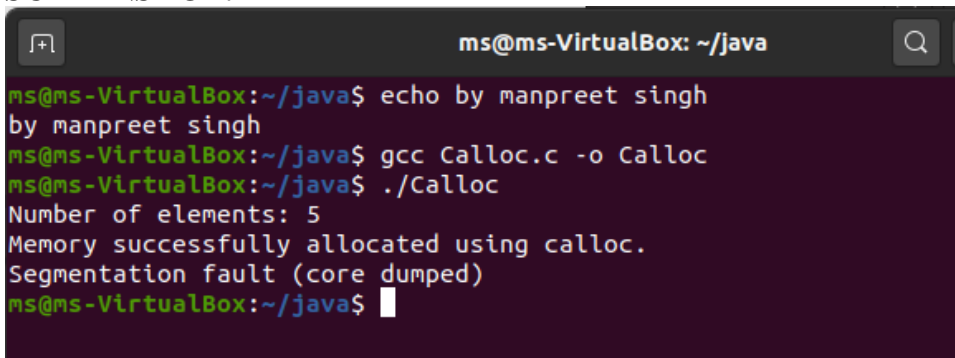
// allocated by calloc or not if (ptr == NULL) {

printf("Memory not allocated.\n"); exit(0);

}
```

```
else {  
  
printf("Memory successfully allocated using calloc.\n");  
  
// Get the elements of the array for (i = 0; i < n; ++i) {  
  
ptr[i] = i + 1;  
  
}  
  
// Print the elements of the array printf("The elements of the array are: "); for (i = 0; i < n;  
++i) {  
  
printf("%d, ", ptr[i]);  
  
}  
  
}  
  
return 0;  
  
}
```

SCREENSHOT:



```
ms@ms-VirtualBox: ~/java  
ms@ms-VirtualBox:~/java$ echo by manpreet singh  
by manpreet singh  
ms@ms-VirtualBox:~/java$ gcc Calloc.c -o Calloc  
ms@ms-VirtualBox:~/java$ ./Calloc  
Number of elements: 5  
Memory successfully allocated using calloc.  
Segmentation fault (core dumped)  
ms@ms-VirtualBox:~/java$
```

Realloc:

```
#include <stdio.h> #include <stdlib.h> int main()
```

```
{
```

```
int* ptr; int n, i;
```

```
// Get the number of elements for the array n = 5;
```

```
printf("Number of elements: %d\n", n);
```

Manpreet singh (03011604423)

```
// Dynamically allocate memory using calloc() ptr = (int*)calloc(n, sizeof(int));

if (ptr == NULL) { printf("Memory not allocated.\n"); exit(0);

}

else {

// Memory has been successfully allocated printf("Memory successfully allocated using
calloc.\n");

// Get the elements of the array for (i = 0; i < n; ++i) {

ptr[i] = i + 1;

}


// Print the elements of the array printf("The elements of the array are: "); for (i = 0; i < n;
++i) {

printf("%d, ", ptr[i]);

}

n = 10;

printf("\n\nThe new size of the array: %d\n", n);

// Dynamically re-allocate memory using realloc() ptr = realloc(ptr, n * sizeof(int));

// Memory has been successfully allocated

printf("Memory successfully re-allocated using realloc.\n");

// Get the new elements of the array for (i = 5; i < n; ++i) {

ptr[i] = i + 1;

}

// Print the elements of the array printf("The elements of the array are: "); for (i = 0; i < n;
++i) {

printf("%d, ", ptr[i]);

}

}
```

```
free(ptr);
```

```
}
```

```
return 0;
```

```
}
```

Screenshot-

```
ms@ms-VirtualBox: ~/java
70, 32571, 32572, 32573, 32574, 32575, 32576, 32577, 32578, 32579, 32580, 32581,
32582, 32583, 32584, 32585, 32586, 32587, 32588, 32589, 32590, 32591, 32592, 32
593, 32594, 32595, 32596, 32597, 32598, 32599, 32600, 32601, 32602, 32603, 32604
, 32605, 32606, 32607, 32608, 32609, 32610, 32611, 32612, 32613, 32614, 32615, 3
2616, 32617, 32618, 32619, 32620, 32621, 32622, 32623, 32624, 32625, 32626, 3262
7, 32628, 32629, 32630, 32631, 32632, 32633, 32634, 32635, 32636, 32637, 32638,
32639, 32640, 32641, 32642, 32643, 32644, 32645, 32646, 32647, 32648, 32649, 326
50, 32651, 32652, 32653, 32654, 32655, 32656, 32657, 32658, 32659, 32660, 32661,
32662, 32663, 32664, 32665, 32666, 32667, 32668, 32669, 32670, 32671, 32672, 32
673, 32674, 32675, 32676, 32677, 32678, 32679, 32680, 32681, 32682, 32683, 32684
, 32685, 32686, 32687, 32688, 32689, 32690, 32691, 32692, 32693, 32694, 32695, 3
2696, 32697, 32698, 32699, 32700, 32701, 32702, 32703, 32704, 32705, 32706, 3270
7, 32708, 32709, 32710, 32711, 32712, 32713, 32714, 32715, 32716, 32717, 32718,
32719, 32720, 32721, 32722, 32723, 32724, 32725, 32726, 32727, 32728, 32729, 327
30, 32731, 32732, 32733, 32734, 32735, 32736, 32737, 32738, 32739, 32740, 32741,
32742, 32743, 32744, 32745, 32746, 32747, 32748, 32749, 32750, 32751, 32752, 32
753, 32754, 32755, 32756, 32757, 32758, 32759, 32760, 32761, 32762, 32763, 32764
,
The new size of the array: 10
Memory successfully re-allocated using realloc.
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, ms@ms-VirtualBox:~/java$ echo by manpreet singh
by manpreet singh
ms@ms-VirtualBox:~/java$
```

Free:

```
#include <stdio.h> #include <stdlib.h> int main()
```

```
{
```

```
// This pointer will hold the
```

```
// base address of the block created int *ptr, *ptr1;
```

```
int n, i; n = 5;
```

```
printf("Number of elements: %d\n", n);
```

```
// Dynamically allocate memory using malloc() ptr = (int*)malloc(n * sizeof(int));
```

Manpreet singh (03011604423)

```
// Dynamically allocate memory using calloc() ptr1 = (int*)calloc(n, sizeof(int));

// Check if the memory has been successfully
// allocated by malloc or not
if (ptr == NULL || ptr1 == NULL) { printf("Memory not allocated.\n"); exit(0);
}
else {
// Memory has been successfully allocated printf("Memory successfully allocated using
malloc.\n");
// Free the memory free(ptr);
printf("Malloc Memory successfully freed.\n");
// Memory has been successfully allocated printf("\nMemory successfully allocated using
calloc.\n");
// Free the memory free(ptr1);
printf("Calloc Memory successfully freed.\n");
}

return 0;
}#include <stdio.h> #include <stdlib.h> int main()
{
// This pointer will hold the
// base address of the block created int *ptr, *ptr1;
int n, i; n = 5;

printf("Number of elements: %d\n", n);
// Dynamically allocate memory using malloc() ptr = (int*)malloc(n * sizeof(int));

Manpreet singh (03011604423)
```

```
// Dynamically allocate memory using calloc() ptr1 = (int*)calloc(n, sizeof(int));

// Check if the memory has been successfully

// allocated by malloc or not

if (ptr == NULL || ptr1 == NULL) { printf("Memory not allocated.\n"); exit(0);

}

else {

// Memory has been successfully allocated printf("Memory successfully allocated using

malloc.\n");

// Free the memory free(ptr);

printf("Malloc Memory successfully freed.\n");

// Memory has been successfully allocated printf("\nMemory successfully allocated using

calloc.\n");

// Free the memory

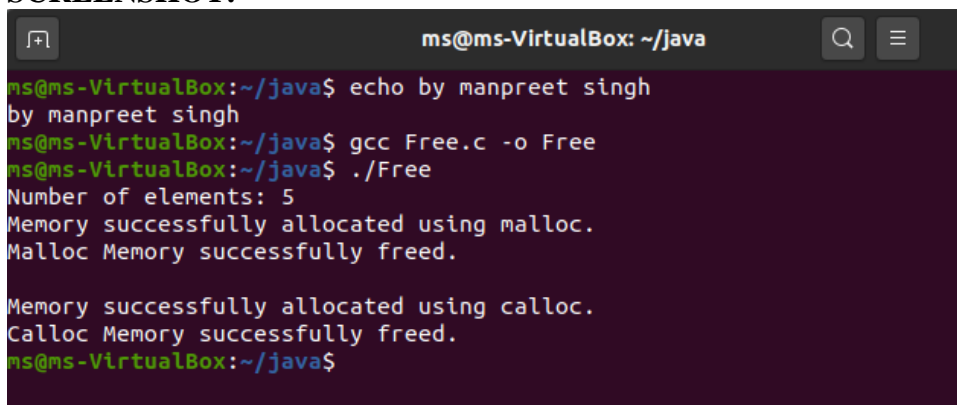
free(ptr1);

printf("Calloc Memory successfully freed.\n");

}

return 0;

}
```

SCREENSHOT:

```
ms@ms-VirtualBox: ~/java
ms@ms-VirtualBox:~/java$ echo by manpreet singh
by manpreet singh
ms@ms-VirtualBox:~/java$ gcc Free.c -o Free
ms@ms-VirtualBox:~/java$ ./Free
Number of elements: 5
Memory successfully allocated using malloc.
Malloc Memory successfully freed.

Memory successfully allocated using calloc.
Calloc Memory successfully freed.
ms@ms-VirtualBox:~/java$
```


P25. Create a text file, named as „courses.txt“ that contains the following four lines:

Java Programming Operating System Discrete Structure

Write a „C“ program that forks three other processes. After forking, the parent process goes into wait state and waits for the children to finish their execution. Each child process reads a line from the „course.txt“ file (Child 1 Reads Line 1, Child 2 Reads Line 2, and Child 3 Reads Line 3) and each prints the respective line. The lines can be printed in any order.

SOLUTION:

```
#include<fcntl.h> #include<unistd.h> #include<sys/stat.h> #include<stdlib.h>

#include<sys/wait.h> #include<stdio.h> void main()

{

pid_t pid; int fd;

int linecount[3]={ 16,17,21 }; int startpoint[3]={ 0,17,33 }; for(int i=0;i<3;i++)

{

pid=fork(); if(pid==0)

{

char read_value[linecount[i]]; fd=open("Courses.txt",O_RDWR);

lseek(fd,startpoint[i],SEEK_SET); read(fd, read_value,linecount[i]);

printf("%s\n",read_value); close(fd);

exit(0);

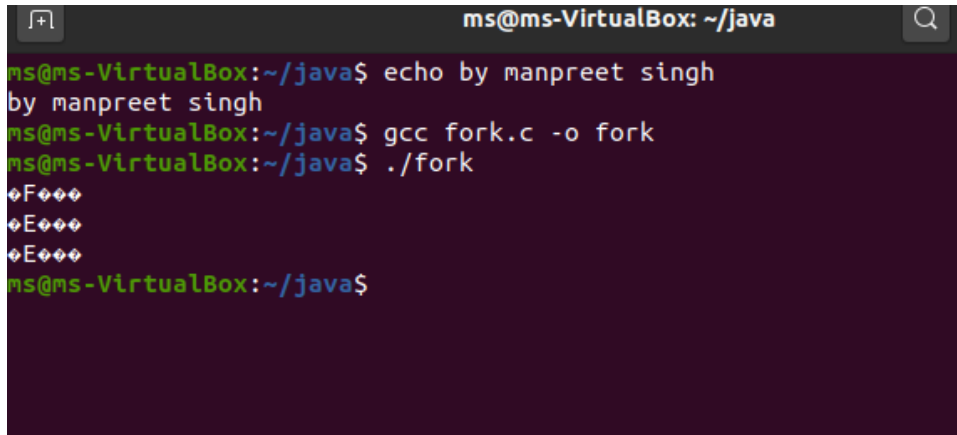
}

else

{

wait(NULL);
```

```
}  
  
}
```

SCREENSHOT:

```
ms@ms-VirtualBox:~/java$ echo by manpreet singh
by manpreet singh
ms@ms-VirtualBox:~/java$ gcc fork.c -o fork
ms@ms-VirtualBox:~/java$ ./fork
♦F♦♦♦
♦E♦♦♦
♦E♦♦♦
ms@ms-VirtualBox:~/java$
```

P26. Write a „C“ program (using appropriate system calls of Linux) that generates „n“ integers and stores them in a text file, named as „All.txt“. Then, retrieve the stored integers from this file and copy to “Odd.txt” and

„Even.txt“ based upon the type of number, i.e. if the retrieved integer is odd number then store in „Odd.txt“ file or if the retrieved integer is even then store in „Even.txt“ file. Finally, display the contents of all three files on the screen.

SOLUTION:

```
#include<stdio.h> #include<fcntl.h> #include<unistd.h> #include<sys/stat.h> void main()
```

```
int n;
```

```
printf("\nUp to How Many Numbers: "); scanf("%d",&n);
```

```
char write_value[n], read_value[n]; int fd,fd_odd,fd_even; printf("\nWriting To All.txt: ");
```

```
for(int i=0;i<n;i++)
```

```
{
```

```
write_value[i]=(i+1)+'0'; printf("%c",write_value[i]);
```

```
}

printf("\n");

fd=open("All.txt",O_CREAT | O_WRONLY, 0777); write(fd,write_value,n);

close(fd);

fd=open("All.txt", O_RDWR); read(fd,read_value,n);

char write_odd[n/2],write_even[n/2]; int ind_odd=0,ind_even=0;

for(int i=0;i<n;i++)

{

int temp=read_value[i]; if(temp%2==0)

write_even[ind_even++]=read_value[i];

}

else

{

write_odd[ind_odd++]=read_value[i];

}

}

printf("\nWriting To odd.txt:"); for(int j=0;j<n/2;j++)

{

printf("%c",write_odd[j]);

}

printf("\n");

printf("\nWriting To even.txt:"); for(int j=0;j<n/2;j++)

{

printf("%c",write_even[j]);

}

}
```

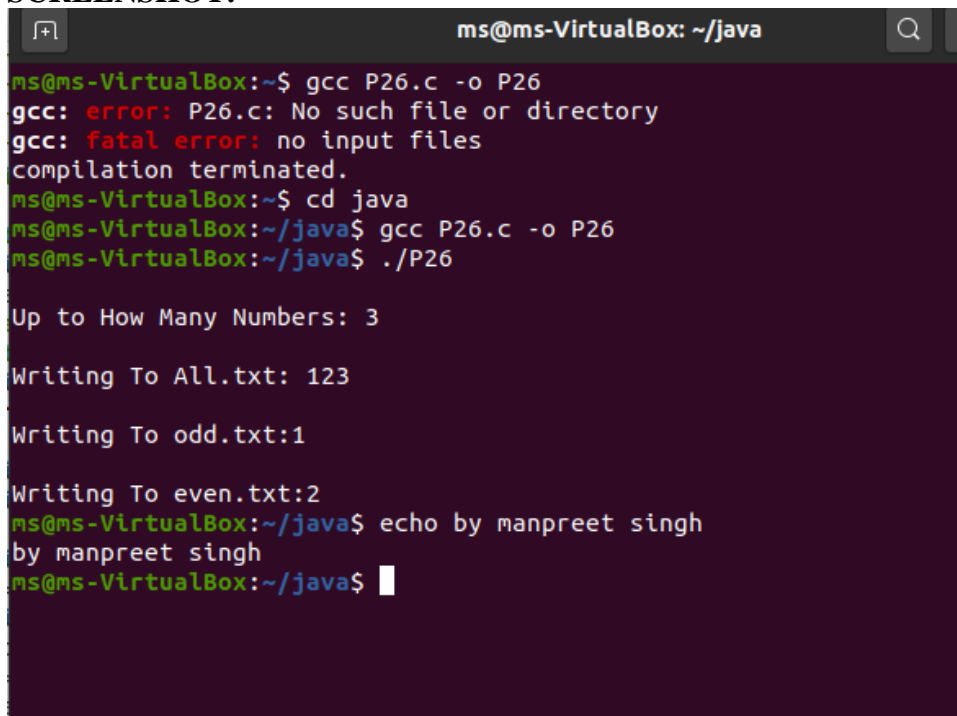
```
printf("\n");

fd_odd=open("odd.txt",O_CREAT | O_RDWR, 0777); write(fd_odd, write_odd, n/2);

close(fd_odd); fd_even=open("even.txt",O_CREAT | O_RDWR, 0777); write(fd_even,

write_even, n/2); close(fd_even);

}
```

SCREENSHOT:

```
ms@ms-VirtualBox: ~/java
ms@ms-VirtualBox:~$ gcc P26.c -o P26
gcc: error: P26.c: No such file or directory
gcc: fatal error: no input files
compilation terminated.
ms@ms-VirtualBox:~$ cd java
ms@ms-VirtualBox:~/java$ gcc P26.c -o P26
ms@ms-VirtualBox:~/java$ ./P26

Up to How Many Numbers: 3

Writing To All.txt: 123

Writing To odd.txt:1

Writing To even.txt:2
ms@ms-VirtualBox:~/java$ echo by manpreet singh
by manpreet singh
ms@ms-VirtualBox:~/java$
```

P27. Write a program in „C“ which accepts the file or directory name and permission (access rights) from the user and then changes the access rights accordingly. Use appropriate system call(s) of Linux.

SOLUTION:

```
#include<stdio.h> #include<fcntl.h> #include<unistd.h> #include<sys/stat.h> int main(){

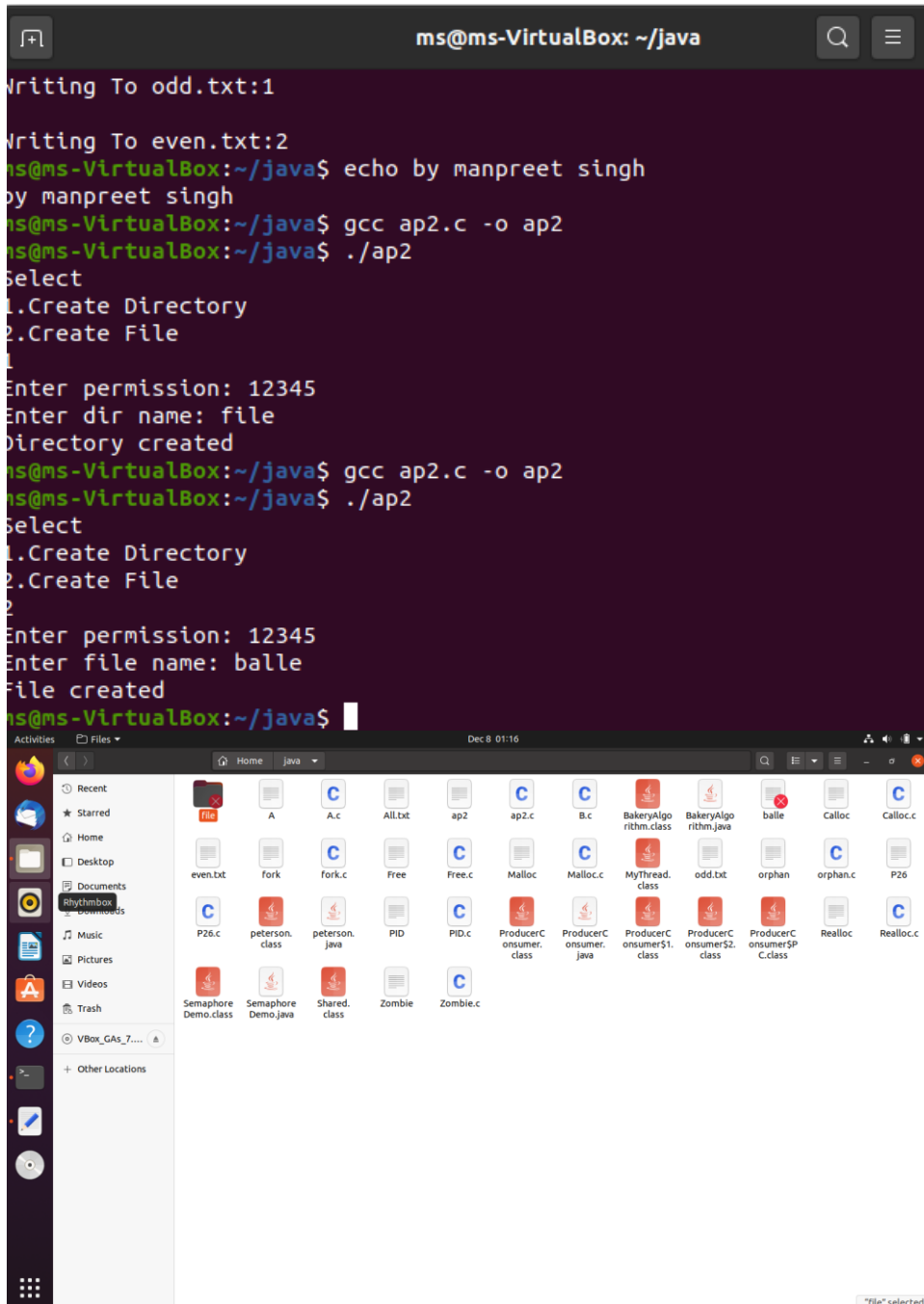
int inp,check,fd; unsigned int perm; char name[51] = ""; printf("Select\n1.Create

Directory\n2.Create File\n"); scanf("%d", &inp);

printf("Enter permission: "); scanf("%o", &perm); switch(inp)
```

```
{  
  
case 1: printf("Enter dir name: "); scanf("%50s", name);  
  
check = mkdir(name, perm); if (!check)  
  
{  
  
printf("Directory created\n");  
  
}  
  
else  
  
{  
  
printf("Unable to create directory\n");  
  
}  
  
break;  
  
case 2: printf("Enter file name: "); scanf("%50s", name);  
  
fd = open(name,O_CREAT | O_RDWR, perm); printf("File created\n");  
  
close(fd);  
  
break;  
  
}  
  
}
```

Screenshot



```
ms@ms-VirtualBox: ~/java
Writing To odd.txt:1
Writing To even.txt:2
ms@ms-VirtualBox:~/java$ echo by manpreet singh
by manpreet singh
ms@ms-VirtualBox:~/java$ gcc ap2.c -o ap2
ms@ms-VirtualBox:~/java$ ./ap2
Select
1.Create Directory
2.Create File
1
Enter permission: 12345
Enter dir name: file
Directory created
ms@ms-VirtualBox:~/java$ gcc ap2.c -o ap2
ms@ms-VirtualBox:~/java$ ./ap2
Select
1.Create Directory
2.Create File
2
Enter permission: 12345
Enter file name: balle
File created
ms@ms-VirtualBox:~/java$
```

The file manager window shows a directory listing of files and folders in the ~/java directory. The files include: even.txt, fork, fork.c, Free, Free.c, Malloc, Malloc.c, MyThread.class, odd.txt, orphan, orphan.c, P26, P26.c, peterson.class, peterson.java, PID, PID.c, ProducerC onsumer.class, ProducerC onsumer.java, ProducerC onsumer\$1.class, ProducerC onsumer\$2.class, ProducerC onsumer\$P C.class, Semaphore Demo.class, Semaphore Demo.java, Shared.class, Zombie, and Zombie.c.

P28. Write a „C“ program (using appropriate system calls of Linux) which generates and stores the characters from „a“ to „z“. Then, display the stored characters in alternative manner, like: a, c, e, g, ..., etc.

SOLUTION:

Manpreet singh (03011604423)

```
#include<stdio.h> #include<fcntl.h> #include<unistd.h>
```

```
#include<sys/stat.h> void main()
```

```
{
```

```
int n=26,ind=0;
```

```
int fd,sizeRead,sizeWrite;
```

```
char write_value[n], read_value[n]; for(int i=65;i<91;i++)
```

```
{
```

```
write_value[ind]=i; ind++;
```

```
}
```

```
fd=open("alpha.txt",O_CREAT |O_RDWR, 0777); sizeWrite=write(fd, write_value,n);
```

```
close(fd); fd=open("alpha.txt",O_RDWR); sizeRead=read(fd,read_value,n); close(fd);
```

```
for(int i=0;i<n;i+=2)
```

```
{
```

```
printf("%c",read_value[i]);
```

```
}
```

```
printf("\n");
```

```
}
```

SCREENSHOT:

```

ms@ms-VirtualBox: ~/java
Enter dir name: file
Directory created
ms@ms-VirtualBox:~/java$ gcc ap2.c -o ap2
ms@ms-VirtualBox:~/java$ ./ap2
Select
1.Create Directory
2.Create File
2
Enter permission: 12345
Enter file name: balle
File created
ms@ms-VirtualBox:~/java$ gcc P28.c -o P28
P28.c: In function 'main':
P28.c:15:3: error: expected ';' before 'fd'
   15 | 43
      |   ^
      |   ;
.....
   19 | fd=open("alpha.txt",O_CREAT |O_RDWR, 0777);
      |   ~~
ms@ms-VirtualBox:~/java$ gcc P28.c -o P28
ms@ms-VirtualBox:~/java$ ./P28
ACEGIKMOQSUY
ms@ms-VirtualBox:~/java$

```

P29. Write a „C“ program (using appropriate system calls of Linux) which receives roll number and names of „n“ students, from the user one-by-one and then stores them in a text file, named as „Student.txt“. After inserting all „n“ roll numbers and names, display the contents of file. Also, display the access rights of the file „Student.txt“.

SOLUTION:

```

#include <stdio.h> #include <stdlib.h> struct student {
int rollno;
char name[200];
};
int writef(FILE *file, struct student *s) { int out = fprintf(
file, "%d\n%s\n",
s->rollno, s->name
);
fflush(file); return out;

```



```
}

int readf(FILE *file, struct student *s) {

if (fscanf(file, "%d", &s->rollno) == EOF) return 0; if (fscanf(file, "%s", s->name) == EOF)

return 0; return 1;

}

int main() { int n;

printf("Enter the number of students: "); scanf("%d", &n);

struct student s;

FILE *file = fopen("Student.txt", "w"); for (int i = 0; i < n; i++) {

printf("Enter the roll no of student %d: ", i); scanf("%d", &s.rollno);

printf("Enter the name of student %d: ", i); scanf("%s", s.name);

writef(file, &s);

}

fclose(file);

file = fopen("Student.txt", "r"); int i = 0;

while (readf(file, &s)) {

printf("Student #%d: %s\n", s.rollno, s.name); if(i == 5) break;

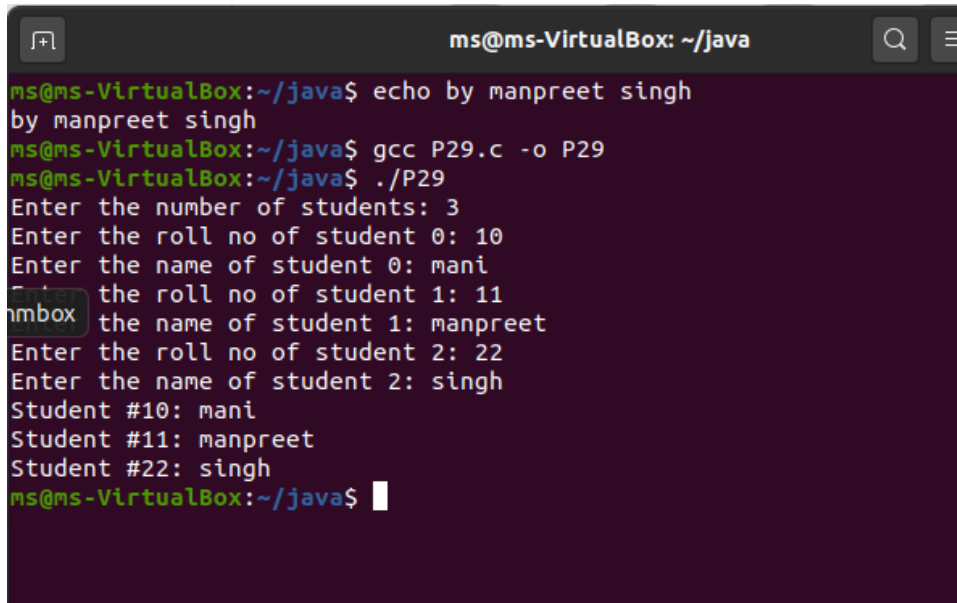
i++;

}

fclose(file); return 0;

}
```

SCREENSHOT:



```
ms@ms-VirtualBox: ~/java
ms@ms-VirtualBox:~/java$ echo by manpreet singh
by manpreet singh
ms@ms-VirtualBox:~/java$ gcc P29.c -o P29
ms@ms-VirtualBox:~/java$ ./P29
Enter the number of students: 3
Enter the roll no of student 0: 10
Enter the name of student 0: mani
Enter the roll no of student 1: 11
Enter the name of student 1: manpreet
Enter the roll no of student 2: 22
Enter the name of student 2: singh
Student #10: mani
Student #11: manpreet
Student #22: singh
ms@ms-VirtualBox:~/java$
```

P30. Demonstrate the use of following system calls by writing an appropriate „C“ program.

1. lseek()

(b) chmod()

2. umask()

(d) access()

3. utime()

a)

SOLUTION

```
#include<unistd.h> #include<stdio.h> #include<fcntl.h> int main()
{
int fd;
char buffer[80];
char msg[50]="Hello i am Anjali."; fd=open("check.txt",O_RDWR); printf("fd=%d",fd);
if(fd!=-1)
{
```

```
printf("\ncheck.txt opened wih read write access\n"); write(fd,msg,sizeof(msg));

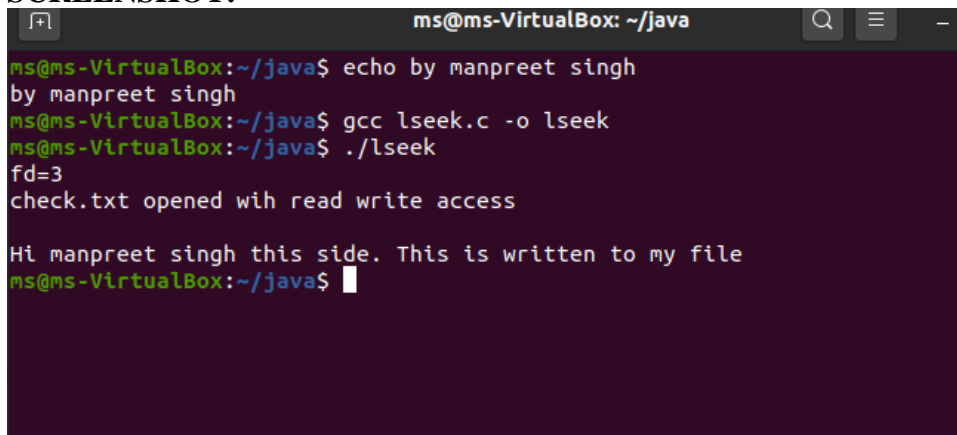
lseek(fd,0,SEEK_SET); read(fd,buffer,sizeof(msg));

printf("\n%s This is written to my file\n",buffer); close(fd);

}

return 0;

}
```

SCREENSHOT:A screenshot of a terminal window titled 'ms@ms-VirtualBox: ~/java'. The terminal shows the following commands and output:

```
ms@ms-VirtualBox:~/java$ echo by manpreet singh
by manpreet singh
ms@ms-VirtualBox:~/java$ gcc lseek.c -o lseek
ms@ms-VirtualBox:~/java$ ./lseek
fd=3
check.txt opened wih read write access

Hi manpreet singh this side. This is written to my file
ms@ms-VirtualBox:~/java$
```

Lseek() (C System Call): lseek is a system call that is used to change the location of the read/write pointer of a file descriptor. The location can be set either in absolute or relative terms.

b. SOLUTION:

```
#include <stdio.h> #include <stdlib.h> #include <string.h> #include <errno.h> #include
<sys/stat.h>

int main(int argc,char **argv)

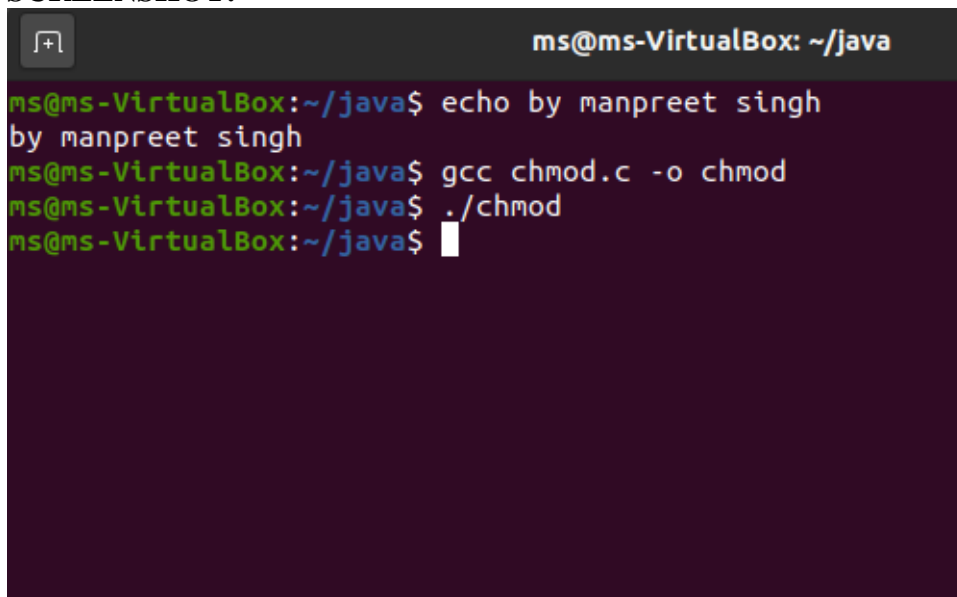
{

char mode[]="0777";

char buf[100]="/home/anjali/First.c"; int i;

i=strtol(mode,0,8); if(chmod(buf,i)<0)
```

```
{  
  
fprintf(stderr,"%s:error in chmod(%s,%s)-%d(%s)\n",  
argv[0],buf,mode,errno,strerror(errno));  
  
exit(1);  
  
}  
  
return(0);  
  
}
```

SCREENSHOT:

```
ms@ms-VirtualBox: ~/java  
ms@ms-VirtualBox:~/java$ echo by manpreet singh  
by manpreet singh  
ms@ms-VirtualBox:~/java$ gcc chmod.c -o chmod  
ms@ms-VirtualBox:~/java$ ./chmod  
ms@ms-VirtualBox:~/java$
```

chmod(): In Unix-like operating systems, the chmod command is used to change the access mode of a file. The name is an abbreviation of change mode. Syntax : chmod [reference][operator][mode] file... The references are used to distinguish the users to whom the permissions apply i.e. they are list of letters that specifies whom to give permissions. The references are represented by one or more of the following letters: Reference Class Description u owner file's owner g group users who are members of the file's group.

**C
SOLUTION:**

```
#define _POSIX_SOURCE #include <fcntl.h>  
  
#include <sys/stat.h> #include <unistd.h> #include <stdlib.h>
```

```
#undef _POSIX_SOURCE

#include <stdio.h>

int main(){ int fd; mode_t old;

printf("Old mask in: %i\n", old=umask(S_IRWXG)); if((fd = creat("Note.txt",
S_IRWXU|S_IRWXG)) < 0){ perror("creat() error");

}

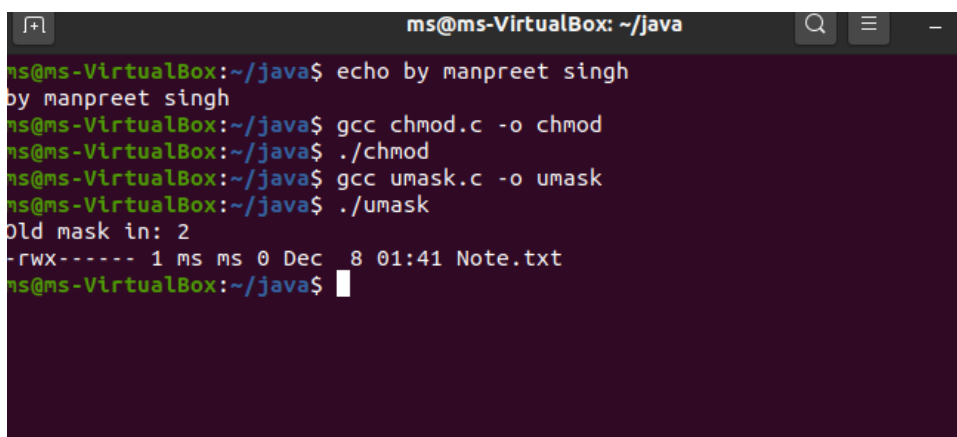
else{

system("ls -l Note.txt"); close(fd); unlink("Note.txt");

}

umask(old);

}
```



```
ms@ms-VirtualBox: ~/java
ms@ms-VirtualBox:~/java$ echo by manpreet singh
by manpreet singh
ms@ms-VirtualBox:~/java$ gcc chmod.c -o chmod
ms@ms-VirtualBox:~/java$ ./chmod
ms@ms-VirtualBox:~/java$ gcc umask.c -o umask
ms@ms-VirtualBox:~/java$ ./umask
Old mask in: 2
-rwx----- 1 ms ms 0 Dec  8 01:41 Note.txt
ms@ms-VirtualBox:~/java$
```

umask() : On Linux and other Unix-like operating systems, new files are created with a default set of permissions. Specifically, a new file's permissions may be restricted in a specific way by applying a permissions "mask" called the umask. The umask command is used to set this mask, or to show you its current value.

b. SOLUTION:

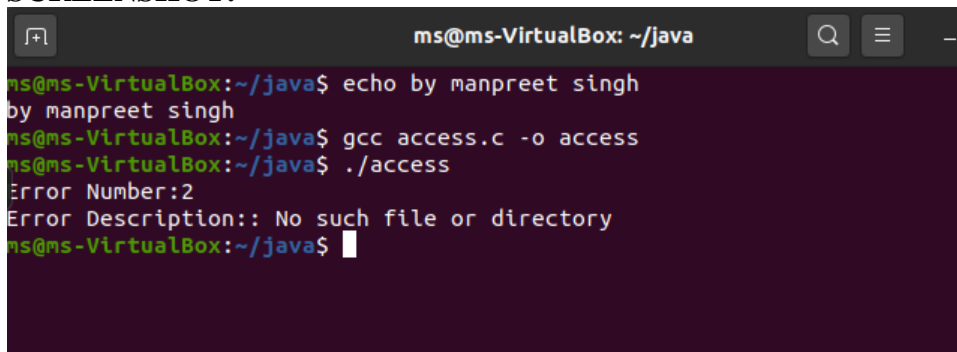
```
#include<stdio.h> #include<unistd.h> #include<errno.h> #include<sys/types.h>
```

```
#include<sys/stat.h> #include<fcntl.h> extern int errno;
```

```
int main(int argc,const char *argv[])
```

Manpreet singh (03011604423)

```
{  
  
int fd=access("Courses.txt",F_OK); if(fd==-1)  
  
{  
  
printf("Error Number:%d\n",errno); perror("Error Description:");  
  
}  
  
else  
  
printf("No error\n"); return 0;  
  
}
```

SCREENSHOT:

```
ms@ms-VirtualBox: ~/java  
ms@ms-VirtualBox:~/java$ echo by manpreet singh  
by manpreet singh  
ms@ms-VirtualBox:~/java$ gcc access.c -o access  
ms@ms-VirtualBox:~/java$ ./access  
Error Number:2  
Error Description:: No such file or directory  
ms@ms-VirtualBox:~/java$
```

access() : In Linux, the access command is used to check whether the calling program has access to a specified file. It can be used to check whether a file exists or not. The check is done using the calling process's real UID and GID.

b. SOLUTION:

```
#include <stdio.h> #include <string.h> #include<stdlib.h>
```

```
int main(void)
```

```
{
```

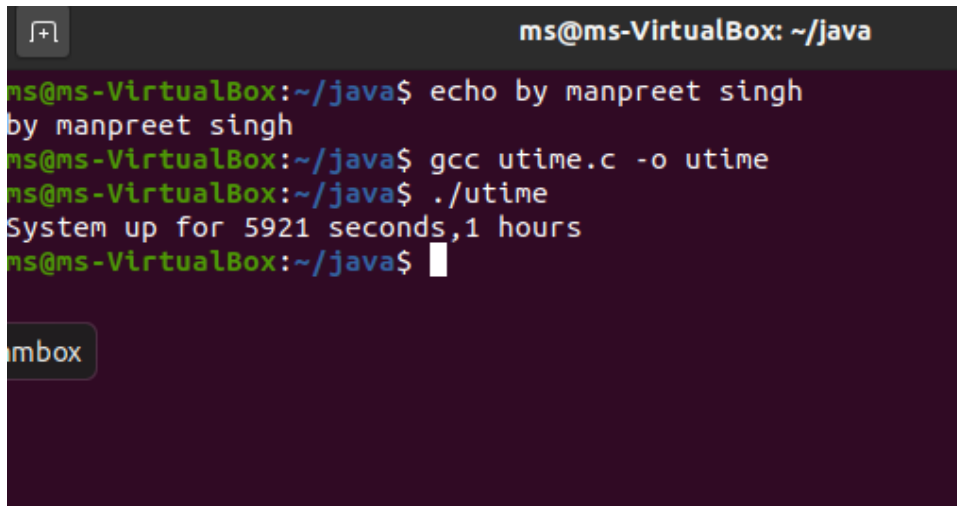
```
FILE* uptimefile; char uptime_chr[28]; long uptime=0;
```

```
if((uptimefile=fopen("/proc/uptime","r"))==NULL) perror("supt"),exit(EXIT_FAILURE);
```

```
fgets(uptime_chr, 12, uptimefile); fclose(uptimefile); uptime=strtol(uptime_chr,NULL,10);
```

Manpreet singh (03011604423)

```
printf("System up for %ld seconds,%ld hours \n",uptime,uptime/ 3600);  
  
exit(EXIT_SUCCESS);  
  
}
```

SCREENSHOT:

The screenshot shows a terminal window titled "ms@ms-VirtualBox: ~/java". The user has entered the following commands and received the following output:

```
ms@ms-VirtualBox:~/java$ echo by manpreet singh  
by manpreet singh  
ms@ms-VirtualBox:~/java$ gcc utime.c -o utime  
ms@ms-VirtualBox:~/java$ ./utime  
System up for 5921 seconds,1 hours  
ms@ms-VirtualBox:~/java$
```

A "Terminal" button is visible in the bottom-left corner of the window.